NASA Contractor Report 191151

On-Board B-ISDN Fast Packet Switching Architectures

Phase II: Development

Proof-of-Concept Architecture Definition Report

Dong-Jye Shyy and Wayne Redman
COMSAT Laboratories
Clarksburg, Maryland

Prepared for
Lewis Research Center
Under Contract NASW–4711

NASA
National Aeronautics and Space Administration

Contents

Contents (Cont'd)

Contents (Cont'd)

Contents (Cont'd)

Illustrations

Illustrations (Cont'd)

Tables

Section 1
Introduction

For the next-generation packet switched communications satellite system with on-board processing and spot-beam operation, a reliable on-board fast packet switch is essential to route packets from different uplink beams to different downlink beams. The rapid emergence of point-to-multipoint services such as video distribution, and the large demand for video conference, distributed data processing, and network management makes the multicast function essential to a fast packet switch (FPS). The satellite's inherent broadcast feature gives the satellite network an advantage over the terrestrial network in providing multicast services. This report evaluates alternate multicast FPS architectures for on-board baseband switching applications and selects a candidate for subsequent breadboard development. Architecture evaluation and selection will be based on the study performed in Phase I and other switch architectures which have become commercially available as large scale integration (LSI) devices.

Although the on-board FPS and the terrestrial ATM switch share many common features and capabilities, the design of an on-board FSP has to consider many additional factors due to the unique satellite communication environment. These factors include mass, power, reliability, fault-tolerance, multicast, switch interfaces, capacity allocation, and congestion control. The design issues for fault tolerance and multicast will be discussed in this report. Congestion control will not be addressed in this report; it will be addressed in the "Critical Element Design and Simulation" task. The design issues for mass, power, reliability, and switch interfaces will be presented in the "High Level Design" task.

The on-board FPS will not only accommodate the ATM cells but also provide services for high-rate/wideband traffic such as the synchronous digital hierarchy (SDH) and synchronous optical network (SONET). The satellite internal packet format adopts satellite virtual packets (SVPs). SVPs are served as a multi-media container to accommodate ATM cells and other types of traffic such as SDH and SONET. The SVP header contains a routing tag, and the on-board FPS routes the SVP to the destination solely based on the routing tag.

This report is organized as follows.

Section 2 presents different multicast FPS architectures and a tradeoff of buffer locations in an FPS for on-board baseband switching applications. A candidate multicast FPS with a proper buffering scheme is selected for detailed investigations and subsequent breadboard development.

Section 3 addresses design considerations for an on-board switching subsystem, including output contention resolution, satellite virtual packet format, priority control, integrated operation of circuit and packet switched traffic, and fault-tolerant design. Specific schemes are proposed to be used in the breadboard design. Based on the

analyses in Sections 2 and 3, high-level functional requirements for the on-board baseband switching subsystems are developed in Section 4.

Section 5 presents a testbed configuration. Section 6 presents preliminary test plans.

Section 2

Alternate Fast Packet Switch Architectures

This section describes the design principles of different multicast switching architectures for on-board baseband switching applications and the selection of the candidate architecture for the breadboard design.

Various multicast switching architectures were extensively discussed in Reference 2-1. Section 2.1 includes a summary of Reference 2-1 Section 6 and other multicast switching architectures proposed recently. Nine multicast switching architectures are described. Section 2.2 addresses the trade-off among different buffering schemes for an FPS. Buffering is necessary for an FPS to temporally store the packets such that output contention and possibly internal blocking can be resolved. Buffering locations not only affect the switch performance, they but also determine the switch complexity. Section 2.3 reviews the switching architectures proposed in recent developments and plans and surveys the commercially available large scale integration (LSI) switch chips for potential space applications. Section 2.4 proposes the selected switching architecture and the queueing scheme for the subsequent breadboard design. Based on various considerations, the self-routing crossbar switch with input buffering is selected.

2.1 Multicast Switching Architectures

A unicast switching architecture has only one major function: routing. A multicast switching architecture must perform two functions: copy and routing. Denote the number of duplications required for a multicast packet as the copy factor (M) .

Some multicast switches implement the two functions using two separate modules: copy module and routing module. The copy module can be implemented using a space-division approach or a time-division approach. In the space-division approach, a copy network is employed to duplicate an exact number of copies of the multicast packet. In the time-division approach, the input port (or the output port) duplicates the multicast packet one by one. The space-division approach has the advantage of fast packet duplication with the price of high hardware cost. The time-division approach has the disadvantage of slow packet duplication, but the hardware cost is minimal. It is possible to combine the space-division and time-division approaches for the copy module to achieve the best compromise among different design considerations.

Others implement the two functions using one common module. These multicast switches have the capability of copying and routing a multicast packet simultaneously.

Based on the above discussion, the multicast switching architectures can be classified into the following four categories:

- space-division copy network plus routing network
- time division copy network plus routing network
- space-division plus time-division copy network plus routing network
- integration of space-division copy network and routing network.

The multicast switching architectures can also be differentiated based on where the packet is duplicated. A multicast packet can be duplicated at a) the input port, b) the switching fabric, and c) the output port. Clearly, for the space-division copy network, a multicast packet is duplicated at the switching fabric. For the time-division copy network, a multicast packet is duplicated either at the input port or at the output port. The above classification is used for the discussion in this report.

This section presents the various multicast switching architectures as follows:

- Input Port Duplication Architecture
  - A.Store-and-forward at the input port
  - B.Store-and-forward using multiple input ports
- Switching Fabric Duplication Architecture
  - C. Copy network plus routing network
  - D. Crossbar switching network
  - E. Sorted-multicast-banyan switching fabric
  - F. Knockout switch
- Output Port Duplication Architecture
  - G. Output port duplication
  - H. Switching fabric duplication plus output port duplication
  - I. Multicast modules at the output port

The selection of a multicast switching architecture for subsequent breadboard design is addressed in Section 2.4.

2.1.1 Input Port Duplication

There are two approaches in this architecture: store-and-forward at the input port and store-and-forward using multiple input ports.

Before these two approaches are described, the concept of "call splitting" is introduced. Call splitting refers to a multicast switch with the capability of splitting a multicast call into multiple calls (either unicast calls or multicast calls). For an input-buffered FPS, there are four ways of routing a multicast packet depending on whether a multicast packet has to be delivered to all the destinations at the same slot or not [2-2].

If the switch has no "call splitting" capability, then the multicast packet must be delivered to all the destinations at the same slot. This is also referred to one shot operation [2-3]. In other words, if at least one of the destinations of the multicast packet can not be delivered at the current slot, the switch must retry to send the multicast packet to the destinations at the next slot time.

The second case is that the switch has a complete call splitting capability, i.e., the transfer of the multicast packet to the destinations can be partially completed. And the point-to-multipoint connection can be completed in S slots, where $1 \leq S \leq \infty$. This type of multicast switches is more flexible and provides better performance (such as throughput and delay). However, the hardware complexity may be increased.

The third one is that the switch has a partial call splitting capability. In this case, the point-to-multipoint connection must be completed in S slots, where $1 \leq S \leq MAXS$. If the multicast connection can not be finished in MAXS slots, the multicast packet is dropped.

The fourth one is to utilize the strengths of the one-shot operation and the complete call splitting capability. The advantage of one-shot operation is to expose the packets behind the head of line (HOL) multicast packet quicker. The advantage of complete call splitting capability is that the output link utilization is higher (compared with no call splitting capability) since the output link will be busy as long as there are packets destined to it. Therefore, there are two steps in the fourth scheme. The first step is to try to send multicast packets to their destinations using the one-shot operation. The next step is to send the remaining multicast packets to their destinations applying the call splitting capability. It is claimed in Reference 2-3 that this scheme achieves the best performance for an FPS with input queueing. The selection for the prototype developed will be determined in the "High-Level Design" task.

A. Store-and-Forward at the Input Port

The arriving packets are stored at the input ports; in other words, the switching architecture employs input queueing. The multicast operation is achieved by duplicating the multicast packet one by one from the input port. The advantage of this approach is that a point-to-point switching fabric can be used for the multicast switching fabric; consequently, the hardware complexity is minimal. The disadvantages are the long packet delay due the serial transfer of a multicast packet and serious congestion if the number of duplication (copy factor) is large. An illustration of the switching architecture is presented in Figure 2-1.

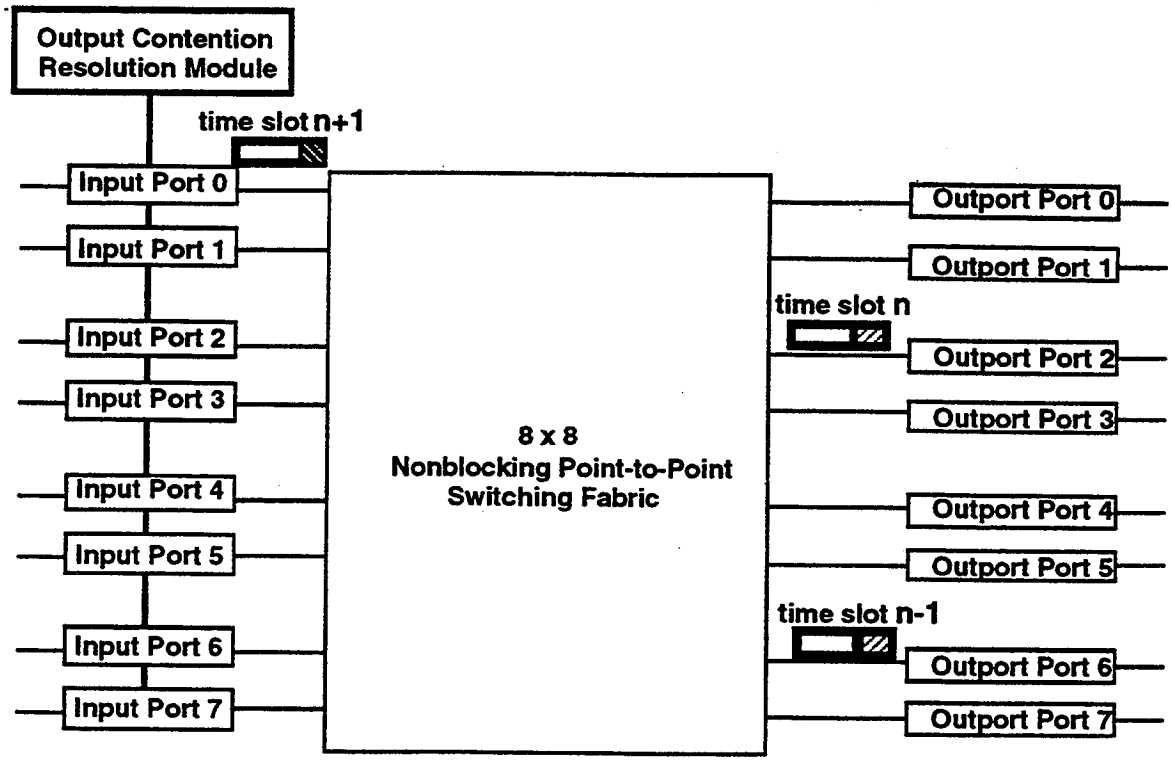


Figure 2-1: Multicast Switching Architecture A: Store-and-Forward at the Input Port

B. Store-and-Forward Using Multiple Input Ports.

In this switching architecture, input queueing is employed. This scheme is an improvement of the previous scheme. In the previous scheme, only one input port is used to duplicate the multicast packet regardless the value of the copy factor. In this scheme, when the copy factor (M) is too large for one input port to handle, multiple input ports are selected by on-board switch controller (OBSC) to duplicate the multicast packet in parallel.

The following presents a procedure of using multiple input ports to duplicate a multicast packet. Designate the input port, which receives the multicast packet, as the primary input port. Since the switching fabric only handles point-to-point connections, the primary input port only can duplicate a single copy of the packet at a time. The primary input port sends a copy of the multicast packet to a designated output port. The output port relays the packet to the corresponding input port (a secondary input port). Since there are two input ports (the primary and the secondary) to handle the copy function, the copy factor for each input port is $\frac{M}{2}$. If the copy factor $\frac{M}{2}$ is still too large for one input port to handle, both primary and secondary input ports repeat the same procedure. After one more iteration, there are one primary and three secondary input ports to handle the copy function, and the copy factor becomes $\frac{M}{2^2}$.

To further improve the delay performance, multiple copies of point-to-point switching fabrics can be stacked in parallel. An illustration of this scheme is shown in Figure 2-2. Parallel copies of switching fabrics not only can improve the throughput performance of the switch, but they also can be used as a copy network. Assume the number of switching fabrics in parallel is 2. The primary input port sends two copies of the multicast packet to two different destinations via two switching fabrics. The two destinations relay the multicast packet to the corresponding input ports (the secondary input ports). Since there are three input ports (one primary and two secondary) to handle the copy function, the copy factor for each input port to handle is $\frac{M}{3}$. If the copy factor $\frac{M}{3}$ is still too large for one input port to handle, the same procedure is repeated. After one more iteration, there are one primary and eight secondary input ports to handle the copy function and the copy factor becomes $\frac{M}{3^2}$.

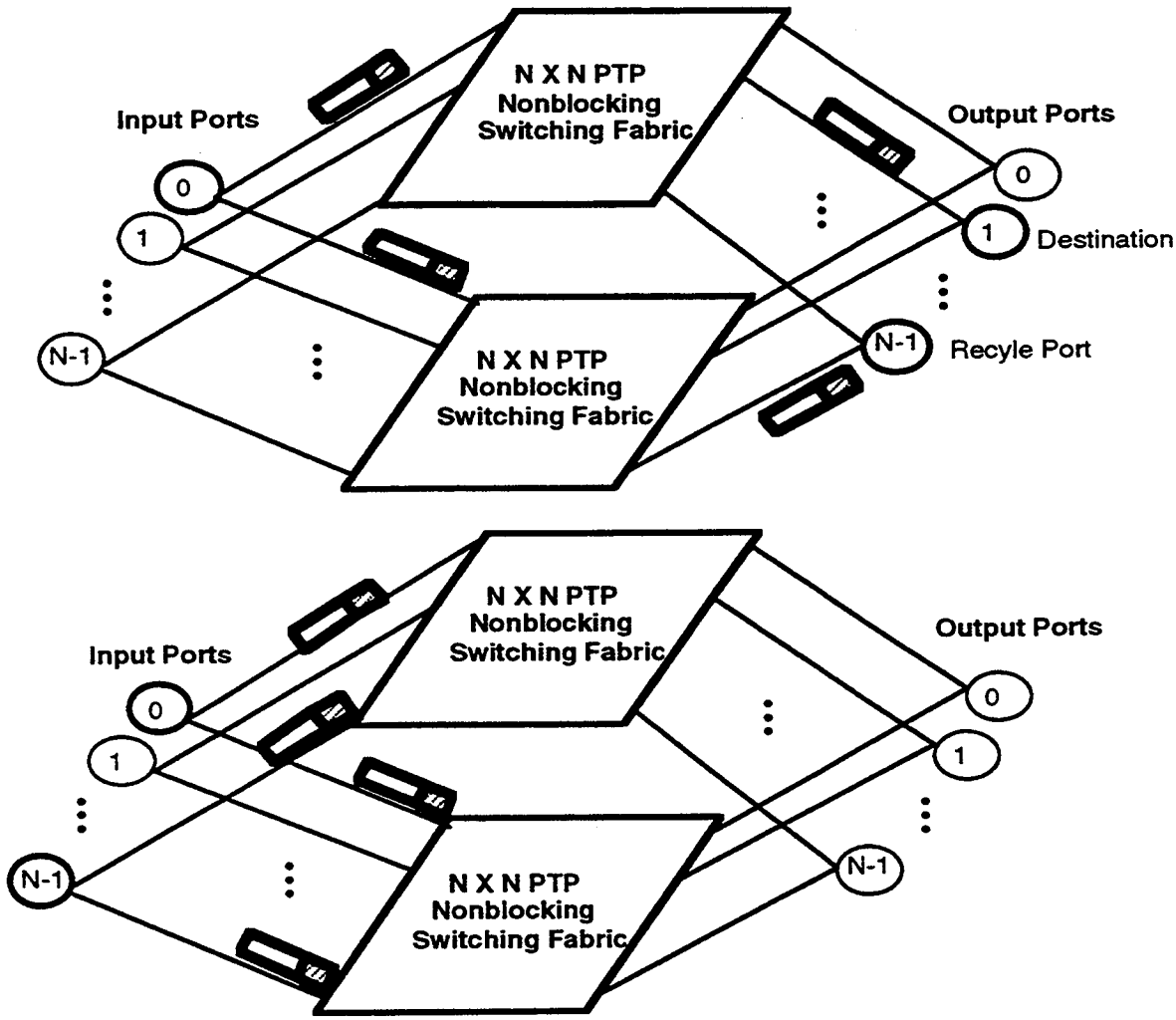


Figure 2-2: Multicast Switching Architecture B: Store-and-Forward using Multiple Inputs with Multiple Switching Fabrics

2.1.2 Switching Fabric Duplication

In this category, the switching fabric is responsible for duplicating and routing the multicast packet. There are two different designs depending on whether the copy and routing functions are separated into two modules or not. Both designs are discussed in detail below.

C. Copy Network Plus Routing Network

Two different copy network designs are discussed in the following.

C.1 Lee's and Turner's approach

In this approach, the arrival packets are duplicated using a space-division copy network [2-4][2-5]. After the packet duplication, the routing network routes the packets to the destinations. Turner and Lee both use this scheme to construct their multicast switches. The difference between the approaches is that the Lee's copy network is nonblocking and the Turner's blocking. Since the Turner's copy network is blocking, the switching elements of the copy network are buffered. The Lee's copy network is superior than the Turner's in several aspects: unbuffered-banyan network, nonblocking property, and constant latency time. The Lee's copy network is used as a representative for discussion.

The function of the copy network is to duplicate an exact number of copies for each multicast packet (see Figure 2-3). This requires the incoming packets to carry a copy factor in the header. For the copy network to be nonblocking, the copy factor must be translated into an address interval. The translation is performed on the incoming packets sequentially from the top to the bottom. The translation is implemented using a running adder network and an address interval encoder. After the packets with proper address intervals are generated, a concentration network concentrates the packets so that the nonblocking condition of the copy network is satisfied. Since these packets address intervals are monotonically increasing (or decreasing) and they are concentrated, a banyan network can duplicate the packets without any blocking. After the copies are generated, a table is necessary to translate the header of each copy to the destination address (routing tag). Buffering is provided in front of the routing network. The routing network routes the packets to the destinations.

There are several disadvantages of using this approach. The first is the delay incurred for every packet (including unicast and multicast packets) passing through the copy network and the routing network. The second is the hardware complexity incurred by the copy network. The third one is the out-of-sequence problem. The duplicated packets may arrive at different input ports of the routing network at different time. If the output contention resolution is resolved before the copy network, i.e., all the copies generated by the copy network have a unique destination, and the routing network is point-to-point nonblocking, there is no out-of-sequence. If the output contention resolution is resolved after the copy network or the routing network is blocking, out-of-sequence may occur.

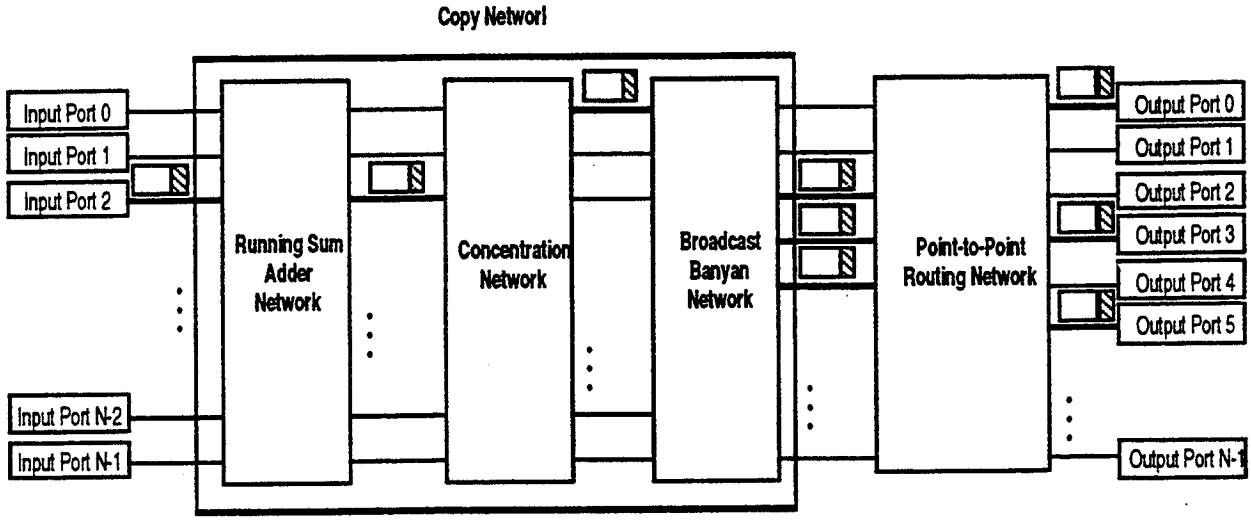


Figure 2-3: Multicast Switching Architecture C.1: Lee's Copy Network Plus Routing Network

C.2 Banyan Copy Network Plus Routing Network

In this special design, a basic banyan network is used as a copy network. There are no running adder network or concentration network as in the Lee's design. The banyan network is operated in time-division multiple access (TDMA) mode. At each TDMA slot, each input port is assigned a set of destinations. The union of these sets of destinations is one permutation pattern of the banyan network. The destination itself is not important. What is important is the number of destinations each input port is assigned. The cardinality of this set is the number of copies which can be made at each slot without any blocking in the banyan network. For unicast packet, the packet passes through the copy network directly. For a multicast packet, the packet is duplicated at each slot to multiple destinations of the copy network. Output contention resolution is required in front of the copy network to guarantee that the number of copies duplicated is always equal to or less than N.

An example is depicted in Figure 2-4. At slot 1, input ports 0 and 2 are assigned destinations 0 and 1. Input ports 1 and 3 are assigned destinations 4 and 5. Input ports 4 and 6 are assigned destinations 2 and 3. Input ports 5 and 7 are assigned destinations 6 and 7. In this scenario, the number of copies which can be duplicated in one slot time is at most 2. To avoid the situations that the input ports with the same destination set always contend with each other, the destination set pattern is changed at every slot following a specific sequence.

The other scenario is that at slot 1, input ports 0 and 2 are assigned destinations 0, 1 , 2, and 3. Input ports 1 and 3 are assigned destinations 4, 5 , 6, and 7. Input ports 4, 5, 6 and 7 have no assignment at this slot time. At slot 2, input ports 4, 5, 6 and 7 have assignment and input ports 0, 1, 2 and 3 have no assignment. In this scenario, the number of copies which can be duplicated in one slot time is at most 4 or 0.

After the copies of the multicast packets are generated, these packets are sent through a routing network to the final destinations. The routing network is a point-to-point nonblocking network. Before the routing network, another stage of output contention resolution is required.

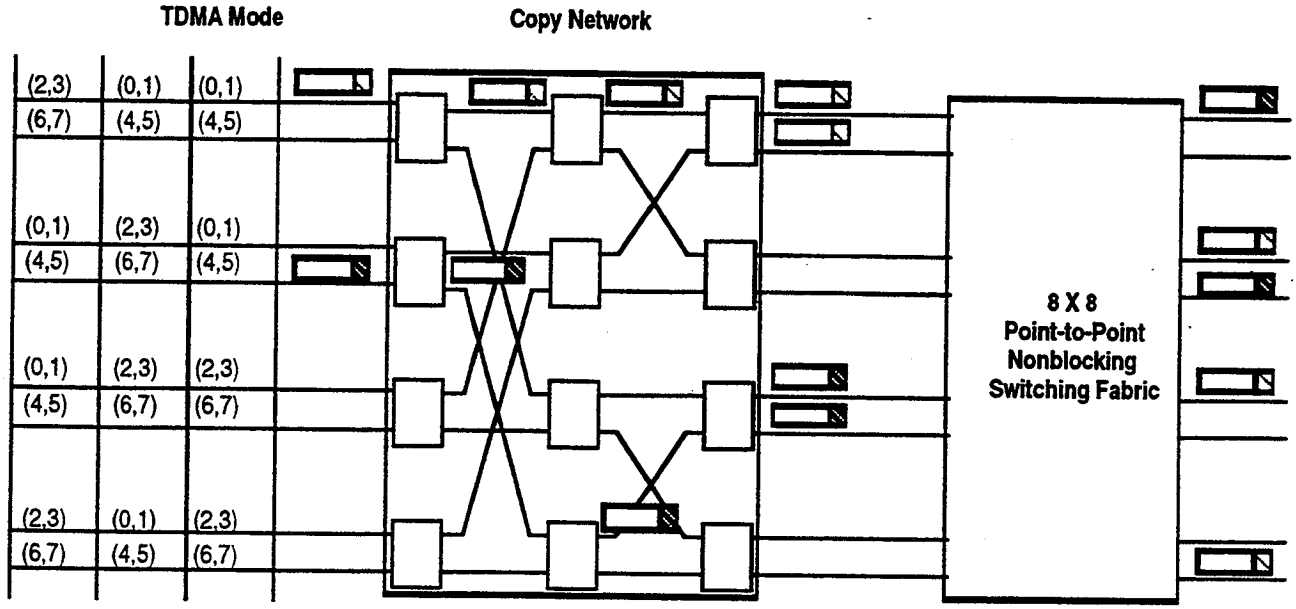


Figure 2-4: Multicast Switching Architecture C.2: Banyan Copy Network Plus Routing Network

D. Crossbar Switching Fabric

Although a crossbar switch has a disadvantage of N^2 growth rate of the number of crosspoints, it has a multicast nonblocking switching fabric. There are two ways of implementing a crossbar switch. The first scheme is to follow the traditional circuit switch design. The crossbar switch is centralized controlled. All the crosspoint states are reconfigured by a central processor. The second scheme is to design a self-routing crossbar (see Figure 2-5). At each crosspoint, an address filter is implemented to extract the packet whose routing tag matches the output port address. Several manufactures have high-speed high-capacity crossbar switches available in the market. They all belong in the circuit switch category. A summary of the switch characteristics is provided in Section 2.2. It may be cost-effective to use the available crossbar switch as a building block and construct a larger multicast switching network.

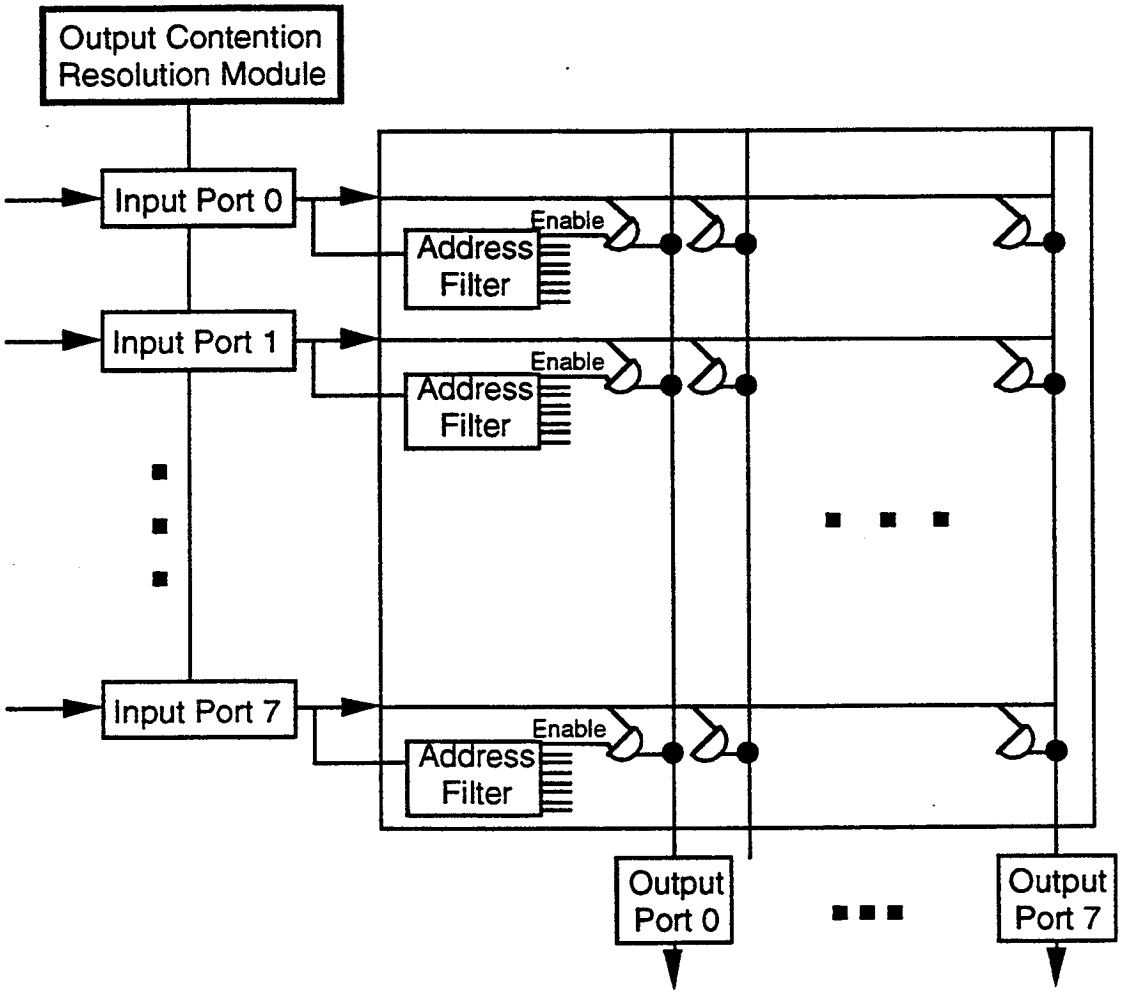


Figure 2-5: Multicast Switching Architecture D: Self-Routing Crossbar

E. Sorted-Multicast-Banyan Switching Fabric

In this category, the switching fabric is capable of duplicating and routing a multicast packet simultaneously. There are two variations in this category depending on the number of sorting networks required in the implementation.

E.1 Cascaded Sorted-Multicast-Banyan Switching Fabric

The switching fabric is based on the multicast banyan network. As in the point-to-point banyan network, the multicast banyan network has internal blocking. It is found that the multicast banyan network can become a nonblocking multicast switching network by using a sorting network in front of every stage of the multicast banyan network [2-9].

Input buffering is used to hold the arriving packets. It is assumed that the input port has the call splitting capability such that the transfer of the packet can be partially completed.

To have a consistent operation of the switching network, empty packets are generated at the input ports if no packets are ready to transmit at a slot time so that the total number of packets at the switching network is always equal to the size of the switch.

The multicast routing field formats use the even and odd group concept associated with the levels of the switching network, and they are arranged using a tree hierarchy structure (see Figure 2-6). The definition of a level in the proposed switching network will be explained later. At level 1, the even group consists of the output addresses whose modulo 2 results are 0; the odd group consists of the output addresses whose modulo 2 results are 1. The addresses at level 1 consist of 2 bits which are used for routing at level 1 of the switching network. There are four possible combinations of the 2-bit format: (1,1), (1,0), (0,1), and (0,0) which represent the destination addresses destined to both groups, even group, empty, and odd group.

The addresses at level 2 consist of 4 bits which are used for routing at level 2. The first 2-bit field is associated with the even group at level 1 and the second 2-bit field is associated with the odd group at level 1. Examine the first 2-bit field. The subeven group within the even group at level 1 consists of the addresses whose module 4 results are 0 and the subodd group within the even group at level 1 consists of the addresses whose module 4 results are 2. Examine the second 2-bit field. The subeven group within the odd group at level 1 consists of the addresses whose module 4 results are 1 and the subodd group within the odd group at level 1 consists of the addresses whose module 4 results are 3.

In general, for a switching network with size N, the addresses at level m consist of 2^m bits, where $1 \leq m \leq \text{Log}_2 N$. The size of the multicast routing tag is 2N - 2.

It can be observed that at stage 1 of the multicast banyan network there is no blocking if only one of the following three situations is allowed to occur at each switching element.

- one packet which destined to both groups and the other packet is an empty packet.
- two packets where one packet is destined to one group and the other is destined to the other group
- one packet which destined to only one group and the other packet is an empty packet.

In order to achieve the above objective, a sorting network is used to rearrange the pattern of the arriving packets. The sorting network sorts the packets using the 2-bit field at level 1. Let the sorting network sort the packets into non-ascending order. After the sorting procedure, the sequence of the packets appears at the outputs of the sorting network is: both groups, even group, empty, and odd group.

Using a shuffle interconnection to connect from the outputs of the sorting networks to the inputs of stage 1 of the banyan network, it is guaranteed that there is no blocking at stage 1 (see Figure 2-7).

It has been shown that there is no blocking at level 1 of the network, where level 1 consists of one sorting network with size N and stage 1 of the banyan network.

The operation of each switching element at stage 1 of the banyan network is described as follows. The switching element routes the packet to the upper link if the 2- bit tag is destined for the even group; it routes the packet to the lower link if the 2-bit tag is destined for the odd group; it routes and copies the packet to both links if the 2-bit tag is destined for two groups. The empty packet is deleted if the other packet at the other input is destined to both groups; otherwise, the empty packet is sent to the next level. In summary, the 2-bit routing bits at level 1 are used for sorting for the N X N sorting network and routing for stage 1 of the banyan network.

After level 1, the packets have been divided into two groups according to the destination routing tags; the packets destined to the even group are routed to the upper subnetwork and the packets destined to the odd group are routed to the lower subnetwork. Level 2 of the routing tag is used for routing at level 2 of the network which consists of two sorting networks with size N/2 in parallel and stage 2 of the banyan network. The upper subnetwork (or the lower subnetwork) consists of one sorting network with size N/2 and the upper half (or the lower half) of stage 2 of the banyan network.

The upper subnetwork with size N/2 uses the first 2 bits at level 2 of the routing tag for routing, and the lower subnetwork with size N/2 uses the second 2 bits at level 2 of the routing tag for routing. The same routing procedure as in level 1 is applied at each subnetwork.

This operation is repeated at every level until the last level. At the last level, the size of each subnetwork is 2. Hence, no sorting network is required in this level. The last level of the network only consists of stage Log_2 N of the banyan network.

The output ports of the switch check the routing tag of the arriving packet to determine it is an empty packet or not. If it is an empty packet, it will be discarded. The logic to perform this operation is very simple, which only needs to check a 2-bit field.

E.2 Sorted-Banyan-Based with Recycling

This approach is a modified version of Architecture E.1. Only one sorting network and one stage of the banyan network is required. However, the sorting network and the routing stage are reused multiple times. In this approach, time is traded with space. In order to reuse the sorting network, the sorting network not only can sort the arrival packets based on their destination addresses, but it also can be reconfigured to sort multiple groups with a smaller size in parallel. For example, an 8 X 8 sorting network can sort a group of size 8 or two groups of size 4 in parallel. At cycle 1, the sorting network sorts a group of size 8. At cycle 2, the sorting network sorts two groups of size 4 in parallel. In general, the multicast function can be finished in Log_2 N cycles.

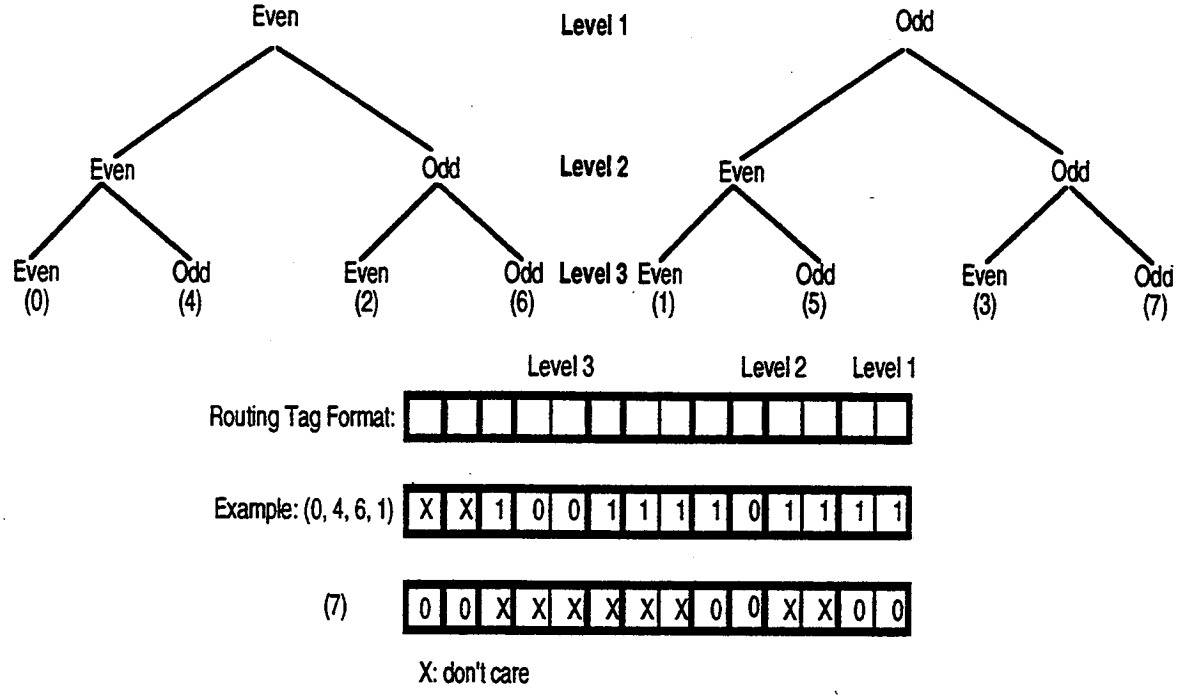


Figure 2-6: Multicast Switching Architecture E.1: Multicast Routing Tag Format

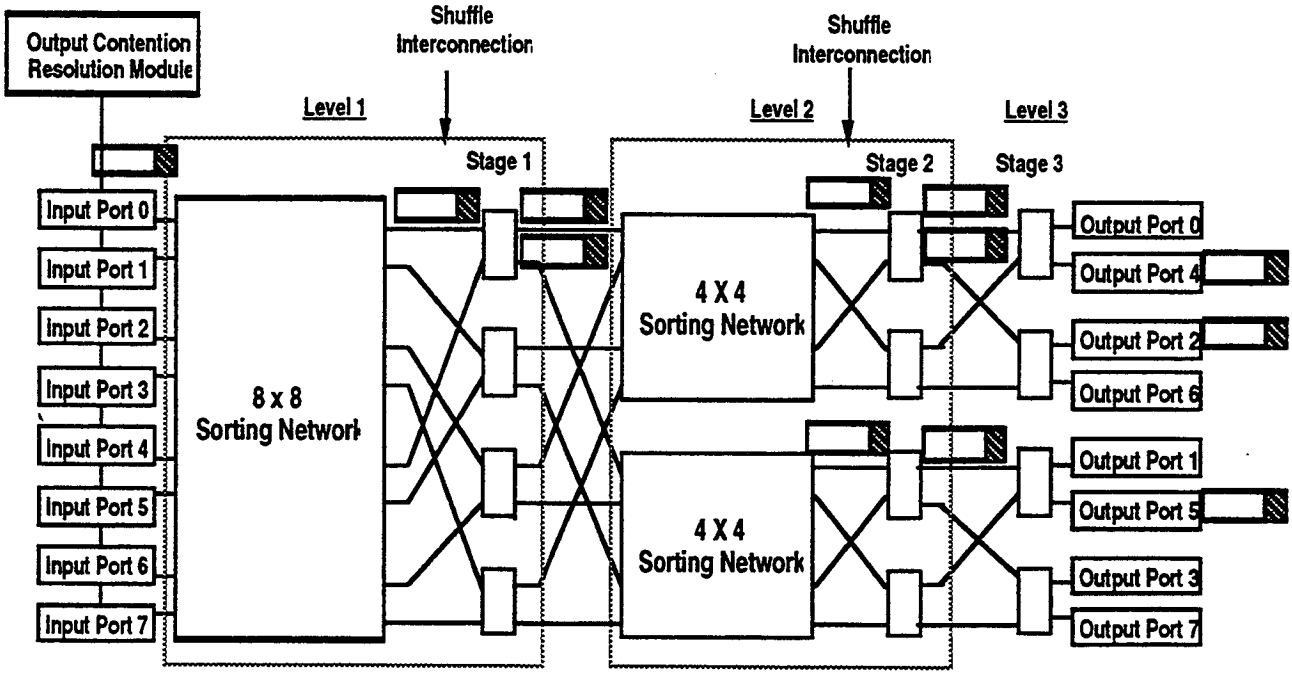


Figure 2-7: Multicast Switching Architecture E.1: Sorted-Multicast-Banyan Switching Fabric

F. Knockout Switch

The knockout switch shown in Figure 2-8 uses the bus approach to interconnect the inputs and outputs [2-10]. There are N broadcast buses, one from each input port, in the switch and there are N filters at each bus interface of the output port. The total number of filters for the switch is N2.

There is a disjoint path between each input and output pair. The switching fabric is point-to-multipoint nonblocking. Since the format of the point-to-point routing tag is different from the point-to-multipoint routing tag, the filter design will be different for both cases. N filters at each output port performs as N receivers which can receive N arriving packets at the same time. After the N receivers, there is one output buffer which performs as a statistical multiplexer. The amount of buffering required at each output port depends on the packet loss ratio requirement.

The main disadvantage of the knockout switch is that the hardware complexity of the output port interface is very high for a large size.

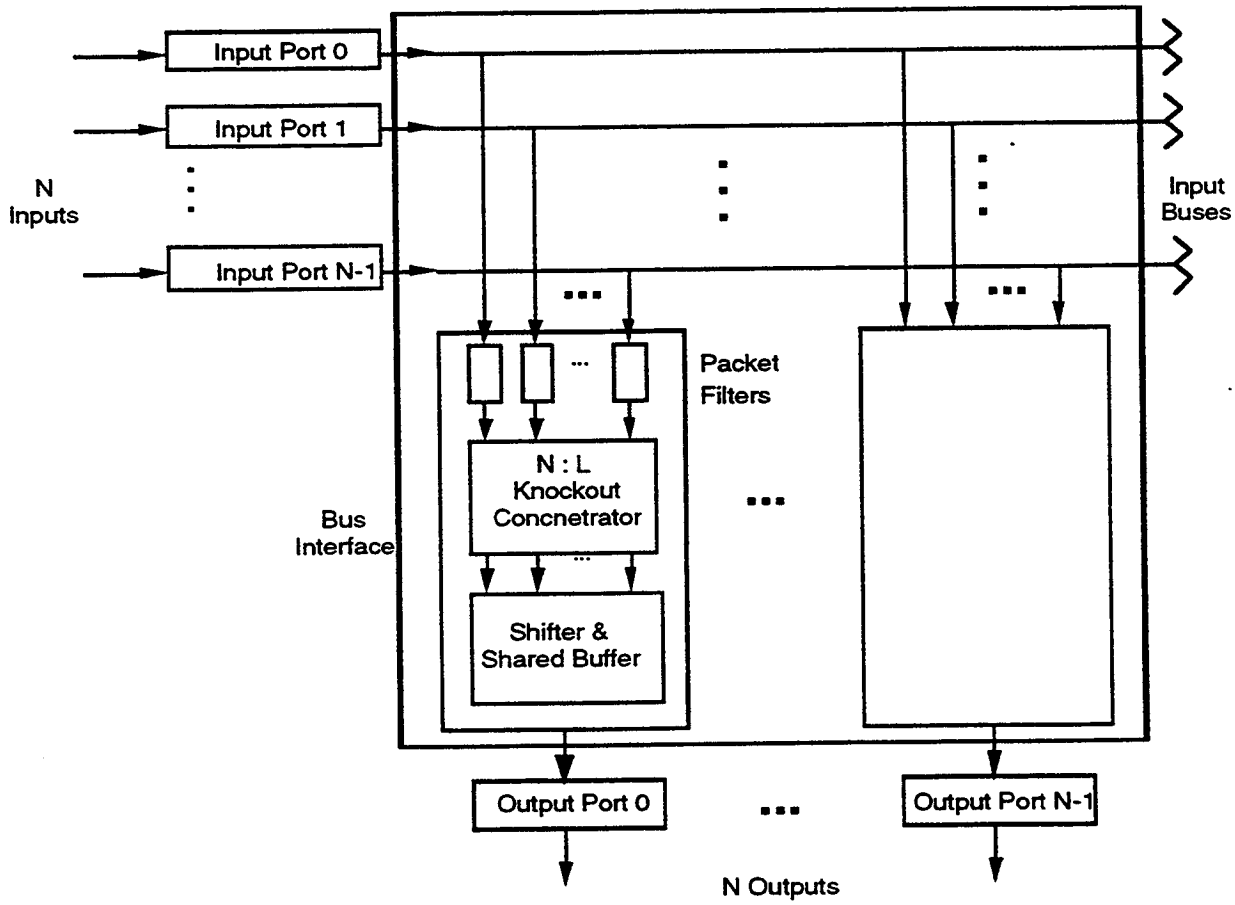


Figure 2-8: Multicast Switching Architecture F: Knockout Switch

2.1.3 Output Port Duplication

G. Output Port Duplication

In this scheme, packet duplication occurs at the output port. Designate the first destination of multiple destinations of the multicast packet as the primary destination. The input port sends the multicast packet to the primary destination first. The output port send the packet to the output line and duplicates one copy of the multicast packet. The copy is recycled back to the corresponding input port. Since the number of copies generated each time is only one, this scheme is equivalent to Architecture A: store-and-forward at the input port.

H. Switching Fabric Duplication Plus Output Port Duplication

The switching fabric comprises buffered switching elements. The switching network topology is a banyan network with extra stages [2-11]. The extra stages are used to duplicate the multicast packet. This switch can duplicate C copies of a multicast packet, and route one of the (C) copies to the destination at one time. The value of C depends on the number of extra stages and the destination pattern of the multicast packet. Clearly one extra stage can generate at most two copies. Two extra stages can generate at most four copies; and so on.

Before describing the multicast operation, the concept of primary destination of a multicast packet is introduced. If multiple destinations are not continuous, any destination can be the primary destination. If some of the destinations are continuous, the first destination in the continuous group is designated as the primary destination. Note the condition that the extra stages can duplicate the packets without blocking is all the destinations are continuous. Therefore, the switching fabric duplicates the packet only when some of the destinations are contiguous. Otherwise, the output port simply duplicates one copy of the multicast packet and recycles the copy back to the corresponding input port.

The input port sends the multicast packet to the primary destination. When the multicast packet passes through the switching fabric and the output port, C copies of the packets are generated. As discussed above, the value of C is a variable. These C copies are recycled back to different input ports. The same procedure repeats until the number of copies generated is equal to the copy factor.

Three examples are given below. Assume the banyan network has two extra stages. Example 1: assume the destinations are (0,1). Then only the last stage duplicates the multicast packet. It takes two cycles to finish transmission of the multicast packet. Example 2: assume the destinations are (0,1,2). The last two stages are used to duplicate the multicast packet. It takes two cycles to finish transmission of the multicast packet. Example 3: assume the destination are (0,3,5). Since the packet destinations are not contiguous, the packet duplication only occurs at the output port. It takes three cycles to finish transmission of the multicast packet.

I. Multicast Modules at the Output Port

In this approach, there are multiple multicast modules at the output ports. All the multicast packets are relayed to these multicast modules first through a point-to-point nonblocking switching fabric. And then the multicast modules send the multicast packet to the destined output ports through a point-to-multipoint nonblocking switching fabric (see Figure 2-9). The number of multicast modules required depends on the amount of multicast traffic.

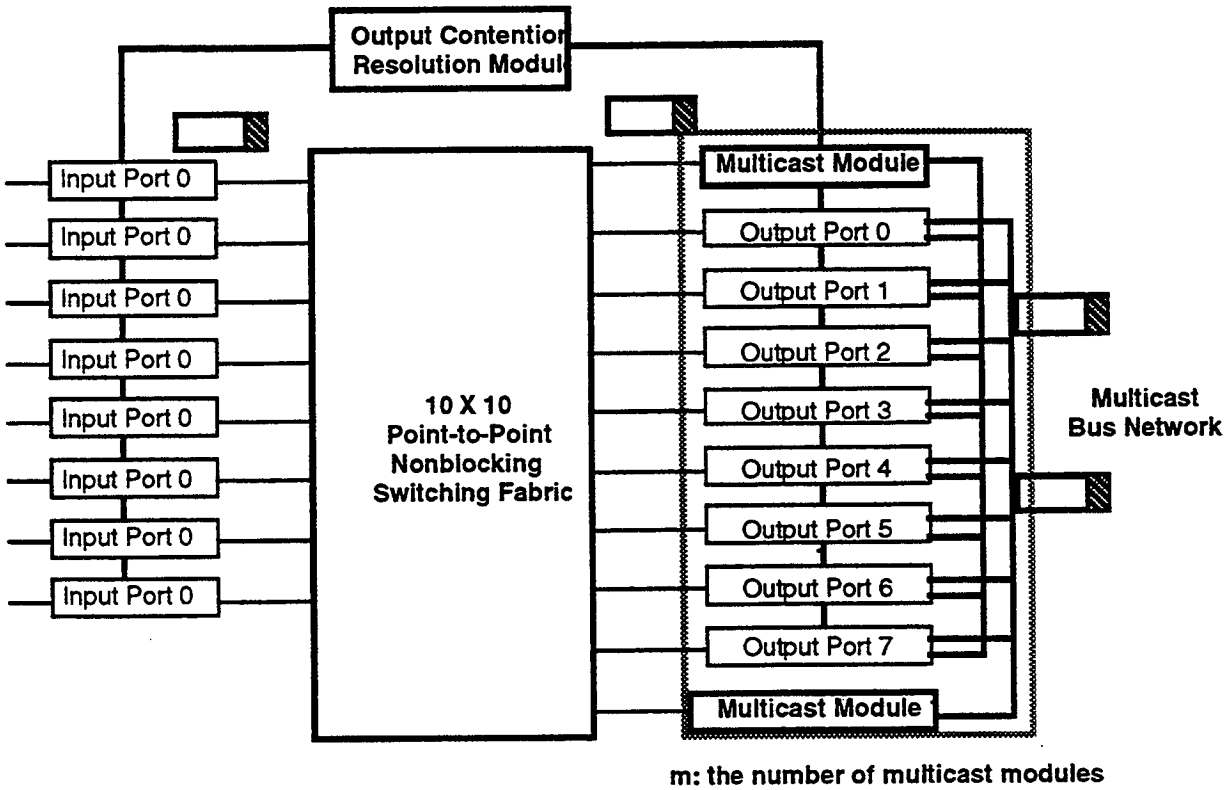


Figure 2-9: Multicast Switching Architecture I: Multicast Modules at the Output Port

The multicast knockout switch uses a similar approach [2-12]. The knockout switch uses a bus to interconnect the inputs and outputs. There are N broadcast buses in the switch for the point-to-point applications. For point-to-multipoint applications, extra multicast modules are required. If there are M multicast modules, then the total number of buses is N + M and the size of the switch becomes N X (N+M). There are (N+M) filters at each bus interface of the output port, where each filter is for one input; hence, the total number of filters for the switch are N^2 + NM. It can be seen that the complexity of the bus interface is very high. The desired point-to-point switching fabric should be a banyan-type network or a crossbar, which is assumed to be the switching fabric in the discussion below. If a banyan-type network or a crossbar is used as the switching fabric, then the number of

filters necessary for the bus interface at each output port is only M, where M is the number of multicast modules.

The output port reservation scheme (such as the centralized ring reservation scheme) is coupled with the multicast module scheme so that the output port reservation scheme can be done not only for point-to-point connections but also for multicast connections. The output port reservation scheme, which is one of the output contention resolution schemes, will be discussed in detail in Section 3.1. The multicast module is treated as one of the input ports by the output reservation module. The start of the token stream alternates among N input ports and m multicast modules. A multicast packet in the multicast module may be destined to several destinations. Among these destinations, some are free and some are busy during the output reservation process. As before, it is assumed that the multicast module has the call splitting capability such that the transfer of the multicast packet can be partially completed.

2.2 Tradeoff of Buffer Locations for a Fast Packet Switch

Buffering is a necessity for a fast packet switch to temporally store the packets such that output contention and possible internal blocking can be resolved. In general, there are three possible buffering locations in a fast packet switch: input buffering, internal buffering and output buffering.

The trade-off among different buffering locations for an FPS is discussed in this subsection. Buffering locations not only affect the switch performance, they but also determine the hardware complexity. The buffering strategy for the proposed FPS candidate, which improves the switch performance without introducing much hardware complexity, is recommended at the end of this subsection.

For the internal buffering approach, the buffers are implemented in every switching element and packets use the store-and-forward scheme to move from one stage to the next stage. The internal buffering approach has the following disadvantages: more hardware, higher queueing delay, out-of-sequence (if alternate paths exist in the switching network), and difficult fault-diagnosis. For the above reasons, the internal buffering approach is not considered for subsequent development.

Due to the statistical nature of packet switching, output contention always occurs in an FPS; as a result, input buffering is necessary (for a multicast crossbar) to schedule the packet transfer and resolve output contention (and possibly internal blocking). To improve the throughput of a switch with input queueing, there are three basic approaches. The first one is to increase the switch speed and the second one is to use parallel switches. The third one is to design a very efficient output contention resolution scheme. Alternate output contention resolution scheme are presented in Section 3.1. For the first two methods, since more than one packet can arrive to the output port at the same time, output buffering is also necessary.

It is possible to create a contention-free switch. A contention-free switch is defined as a switch whose output port can receive up to N packets in one link slot time. For a contention-free switch, only output buffering is necessary. When the speed switch is

increased N times of the link speed and the switching fabric is point-to-multipoint nonblocking, the resulting fast packet switch is contention free. The speedup factor (S) is defined as the ratio of switch speed and link speed. If the number of switching fabric stacked in parallel (P) is N, the number of receivers at the output ports is N and the switching fabric is point-to-multipoint nonblocking, the resulting fast packet switch is also contention free.

Although a contention-free switch eliminates the input queue, the high speed requirement or the high hardware complexity make this method feasible only for a switch with a very small capacity. The contention-free switch will not be considered for the subsequent development. By allowing output contention to occur, the speed requirement and the hardware complexity can be reduced. The possible configurations of the fast packet switch with different S and P are illustrated in Table 2-1.

Table 2-1A: Correspondence Between Buffering Locations and Switch Speedup

switch speedup (S)	S = 1	1 < S < N	S = N
	input buffering	input buffering + output buffering	output buffering

Table 2-1B: Correspondence Between Buffering Locations and Parallel Switches

parallel switch (P)	P = 1	1 < P < N	P = N
	input buffering	input buffering + output buffering	output buffering

2.2.1 Output Queueing

The queueing model for a nonblocking switching fabric with output queueing has been studied in [2-10][2-11]. The derivation of the average output queue length follows that for the M/G/1 queueing system [2-12]. The assumption of the queueing analysis is that the packet arrival process is independent and it follows the identical Bernoulli process. The destination distribution of packets is uniform. Derivation of the queueing equations to obtain the average queueing length, throughput and CLR is provided in the Appendix A of Reference 2-13. Please refer to it for details. The switch throughput can potentially reach 1. However, the queueing delay will be infinite when the switch throughput is close to 1. A desirable throughput for a switch with output queueing should be around 0.9.

2.2.2 Input Queueing

The queueing analysis for a nonblocking switching fabric with input queue and first in first out (FIFO) buffers was researched in [2-10][2-11]. Due to the head of line blocking (HOL) problem, the switch throughput is bounded by 0.586 for a larger N. The saturation throughput and queueing length were derived analytically. A summary of the queueing derivation for the switch throughput is provided in Appendix A.

As previously discussed, besides increasing the switch speed and use parallel switches, the third approach to increase the throughput of the switch is to design a more efficient contention resolution algorithm. One of the possible algorithms is to use non-FIFO input queue with the windowing scheme. More discussion on this subsection is provided in Section 3.1 If the first packet is blocked due to output contention, the scheduling algorithm also examines the packet on the back of the first packet. The number of packets examined each time depends on the preset window size or the "checking depth". The saturation throughput for different switch sizes and checking depths are provided in Table 2-2 [2-11]. The saturation throughput for different checking depths is obtained using simulation.

Table 2-2: Saturation Throughput for Different Switch Sizes and Checking Depths.

N\checking depth	1	2	3	4	5
2	0.75	0.84	0.89	0.92	0.93
4	0.66	0.76	0.81	0.85	0.87
8	0.62	0.72	0.78	0.82	0.85
16	0.60	0.71	0.77	0.81	0.84
32	0.59	0.70	0.76	0.80	0.83
64	0.59	0.70	0.76	0.80	0.83

From the above table, a large checking depth is an effective way of improving the switch throughput. However, the improvement of throughput gets less when the checking depth gets larger. Increasing the switch speed to further increase the throughput may be necessary.

2.2.3 Input Queueing Plus Output Queueing

In general, providing P parallel switches and increasing the switch speed N times faster have exactly the same effect in improving switch performance when P = N. This

result is the well known trade-off between space and time. For this reason, this subsection only discusses the effect of increasing the switch speed. Although the switch speedup S does not have to be an integer, for easy queueing analysis, the switch speedup S is assumed to be an integer. Since the switch operates S times faster than the link speed, in one link slot, there are S switch slots, where one switch slot is defined as the ratio of the packet size and the switch speed. The output port will, at most, receive S packets in one link slot time. Since the output port may receive more than one packet in one link slot time, output queueing is necessary to store the packets. The switch can process the S packets at the input port in one link slot. Now the question is how many packets per input port the switch can send in one link slot time after the switch is operated S times faster. The first solution is to allow the switch to send, at most, one packet at the input queue in one link slot [2-14][2-15][2-16]. That is to say if the HOL packet is sent out at an input port, the next packet in the queue can not be advanced to the HOL position until the next link slot time. The saturation throughput for different values of S is provided in Table 2-3 [2-15].

Table 2-3: Saturation Throughput for Different Values of Speedup Factors

Speedup Factor (S)	Saturation Throughput
1	0.5858
2	0.8845
3	0.9755
4	0.9956
5	0.9993
6	0.9999
∞	1

Using this approach, the correspondence between time (S) and space (P) does not exist any more.

The second solution is that the switch can send from one up to S packets at the input port in one link slot time. The improvement of the switch throughput is proportional to the switch speed. There is a correspondence between time (S) and space (P).

In terms of implementation, both approaches present little difference. However, the improvement of the switch throughput for the first approach is limited. The first approach underutilizes the advantages provided by increasing the switch speed.

The queueing analysis presented in Appendix A is also applicable in this scenario. The system can be modeled as a tandem queue (two queues in series): the virtual queue at the input port and the output queue. The trade-off between the input queueing and output queueing can be discussed in terms of two system performance measurements: the queueing delay and the packet loss ratio. The queueing delay is the sum of the virtual

queueing delay and the output queueing delay. Although the queueing delay is analytically solvable, it is not easy to obtain the queueing equations for the packet loss ratio. The packet loss ratio is the sum of the packet loss ratio for the input queue and the packet loss ratio for the output queue. It is expected that the packet loss ratio must be obtained using simulation. The derivation of the switch throughput when the switch is operated twice faster than the link speed is presented in Appendix A.

From the discussion in [2-10][2-11], the queueing delay and packet loss ratio of the packet switch with output queueing outperforms those of the packet switch with input queueing.

Evidently, the amount of buffer at the input queue and the output queue depends on the switch speed. When the switch speed is low, more buffer should be placed at the input queue. When the switch speed is high, more buffer should be allocated at the output queue [2-16].

2.2.4 Buffer Size Requirement

The buffer requirement for an 8 x 8 contention-free switch with output queueing can be computed using queueing analysis when packets arrivals follow Poisson distribution. Derivation of the packet loss ratio (PLR) as a function of the buffer size is given in Reference 2-13. The results are shown in Figure 2-10. With a buffer size of 100, the PLR is 10^{-9} when the link utilization is 0.92. The maximum achievable throughput for the contention-free switch with output queueing is close to 1.0.

The buffer requirement for an 8 x 8 switch with input queueing when packets arrivals follow Poisson distribution was obtained using simulation approach [2-1]. The buffer size required to achieve a PLR of 10^{-9} is around 100 when the link utilization is 0.55 and the maximum throughput is 0.6.

For a switch with input queueing and output queueing, the buffer size requirement for input port and output port is not symmetric. When the speedup factor is low, the throughput of the switch is not large enough to accommodate the incoming traffic; hence, most packets are accumulated at the input port. Increasing the switch speed, the packets accumulation starts shifting from the input port to the output port. When the speedup factor is high enough, most packets are accumulated at the output port. Note the throughput of the switch is the arrival rate of the output queue. If the speed continues to increase, the input queue length can be made very small. This result suggests that we should increase the switch speed as high as possible to largely increase the throughput so that most packets are accumulated at the output port. It should be noted when the incoming traffic is unbalanced and/or time varying, the throughput of the switch degrades and, as a result, the packets may be accumulated at the input port. The other consideration is that when faults occur in the switch subsystem, the throughput performance degrades, and as a result, packets are also accumulated at the input port. Hence, an adequate size of buffer should always be provided at the input port as a safety margin.

Since a large number of packets are accumulates at the output port, some output buffers may be full and packets will be dropped. The flow control scheme to prevent packet loss at the output port is investigated latter.

From the above discussion, a general guideline to choose the buffer size is that if the Patio of link utilization and throughput is about 0.9, the buffer size required to achieve a CLR of 10^{-9} is about 100 packets. For bursty traffic, the amount of buffer size to achieve the same PLR will be increased. The amount of increase depends on the traffic characteristics (such as burstiness, peak rate and burst length).

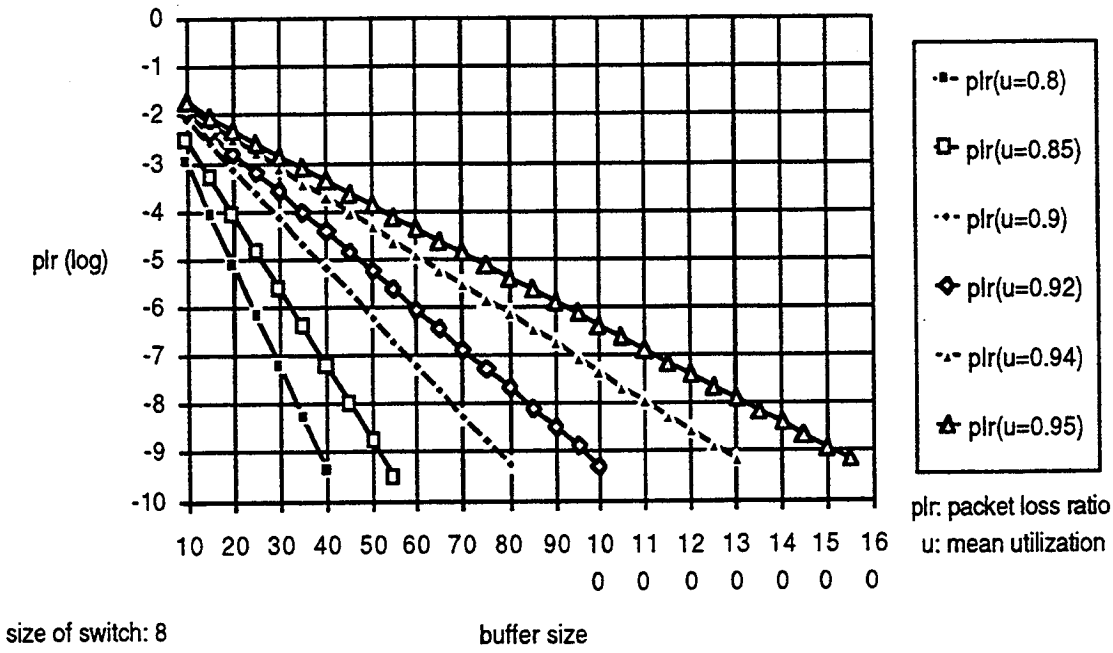


Figure 2-10: PLR vs Buffer Size for an 8 x 8 Contention-Free Switch with Output Queueing

A general comparison table for input queueing, output queueing, input and output queueing, and internal queueing are provided in Table 2-4. This tables compares the following:

- the methods to improve the switch performance
  - larger checking depth
  - switch speedup
  - parallel copies

- larger switching element size

- types of blocking encountered in the switch
- throughput
- ability to perform speed and format conversion
- output contention resolution mechanisms
- fault-diagnosis and hardware complexity

Table 2-4: Comparison of Different Queueing Strategies

	Input Buffering		Output Buffering		Input/Output Buffering	Internal Buffering
	Nonblocking Crossbar or Banyan-Type	Blocking Banyan-Type	Nonblocking Crossbar or Banyan-Type	Knockout Switch	Nonblocking Crossbar or Banyan-Type	
Larger Checking Depth (d)	applicable	applicable (with parallel copies)	no	no	applicable	applicable
Speed Up Factor (S)	$1 \le S < N$ (required output buffering)	$1 \le S \le N$ (required output buffering)	S=N	S=1	$1 < S < N$	$1 \le S \le N$
Multiple parallel copies (P)	applicable	applicable	applicable	no	applicable	applicable (out-of-seq problem)
Larger Switching Element Size (D)	no	applicable	no	no	no	applicable
Head of Line Blocking	yes	yes	no	no	yes	yes
Internal Blocking	no	yes	no	no	no	yes
Output Contention	yes	yes	no	yes	yes	yes
Format Conversion (Speed Conversion)	no	no	yes	yes	yes	no
Output Contention Resolution	• output reservation at input ports • path setup	path setup	contention free	output filter	• output reservation at input ports • path setup	store-and-forward
Throughput	58%	< 58%	100%	100%	≥ 58% and ≤ 100%	≥ 58% (3 buffers or more)
Fault-Diagnosis	easy	easy	easy	easy	easy	hard
Hardware Complexity	low/medium	low	high	high	medium	high

• Throughput and packet transfer delay can be improved by adjusting the value of d, S, P, and/or D

A fast packet switch with input queueing has the least hardware complexity. For on-board processing applications, the input-queued FPS should be chosen. However the throughput of a switch with input queueing is limited at 0.58. There are two recommended approaches. The first one is to design a very efficient scheduling algorithm to largely improve the throughput. Since the switch speed is not increased, output queueing is not necessary. The second one is to use the basic scheduling algorithm (such as centralized ring reservation scheme) and increase the switch speed. The drawback of this scheme is that output queueing is required. More discussion on scheduling (or output contention resolution) is provided in Section 3.1.

2.2.5 Flow Control for a Fast Packet Switch with Input Queueing and Output Queueing

Queueing packets at the input port is required for an FPS with nonblocking switching fabric to resolve output contention, and queueing packets at the output port is necessary if the switch speed is increased to improve the throughput. This subsection discusses the flow control scheme to prevent buffer overflow at the output queue [2-17].

The output contention resolution scheme at the input ports can be implemented with the output port reservation scheme. The output port reservation scheme is assumed to use the centralized ring reservation scheme. In this case, the flow control algorithm is very straightforward. If one output port is congested, no packet is allowed to send to the output port from the input ports. Since the packet has to reserve the output port before transmission begins, by removing the token associated with the congested output port in advance, no input port can request the congested output port. Although the input port can not send any packet to the congested output port, an output port congestion indication may have to send to the input ports so that the situation between an output port is busy (due to contention) and an output port is congested (due to congestion) can be distinguished. The immediate drawback of this scheme is that the HOL blocking problem at the input queue is worsened by the flow control scheme. The reason is that the packet destined to the congested output port will remain in the HOL position for sometime and the result is that the throughput of the switch degrades. One approach is to use a larger checking depth. The other approach is to put the packets destined to the congested output port to a separate queue for temporal storage. There are two separate queues in the input port. The congested queue is used to store the packets whose destination is in congestion. The normal queue is used to store the packets whose destination is not congested.

Since the arrival time of the packets in the congested queue are ahead of the packets in the normal queue, it is reasonable to give priority to the packets in the congested queue over the normal queue. In operation, the input port always check the congested queue first. If the destination of the HOL packet in the congested queue is still in congestion, the packet can not be transmitted because the token associated with the congested output port has been removed. As a result, the input port will check the normal queue. Note with this arrangement, the packet will not be transmitted out-of-sequence.

The interactions between the congested packet and the normal packet is separated using the two-queue scheme. Note that when an input port receives a congestion

notification from the congestion controller, the input port only checks whether the HOL packet in the normal queue is destined to the congested output port or not; the input port does not check the HOL packet in the congested queue.

If the buffer space in the congested queue is full, there are two options: the packets will either stay in the normal queue without transferring over to the congested queue or the packets will be dropped. We may expect that to use the buffer more efficiently, the congested packet should stay in the normal queue if the congested queue is full. However, if the congested packet stays in the normal queue, the switch throughput is degraded due to the HOL blocking problem in the normal queue. Which option is more efficient in reducing the packet loss will be determined using simulation. It is expected that by shifting the congestion between the input buffer and the output buffer, the buffer space can be utilized most efficiently, a short term traffic congestion can be absorbed, and the packet loss can be reduced to a minimum.

The following discussion addresses the necessary modification for multicast traffic. If any one of the destinations of the multicast packet is in congestion, the input port will finish transmitting the multicast packet to different non-congested output ports first. Then the input port will store the multicast packet in the congested queue. It is possible that more than one of the destination output ports are in congestion. To simplify the input port function, the multicast packet will be split into point-to-point packets before being stored in the congested queue. After this, the function of the input port is the same as that in for point-to-point traffic.

2.3 Recent Developments and Plans for ATM Switches

This subsection is to review the switching architectures used in the recent developments of fast packet switching systems/chips for potential on-board applications.

Different terms have been used for referring to the fast packet switching technology. The United States uses fast packet switching, Europe uses asynchronous time division switching, CCITT uses ATM switching, and AT&T uses wideband packet switching.

Since packets are self routed through the FPS, several packets from different input ports may be destined to the same output port at the same time. This situation is referred as output contention. If this occurs, output contention resolution has to be performed such that only one packet is allowed to be transmitted to the output port. The output contention resolution scheme along with the switching architecture must be designed carefully such that the quality of service (QOS), such as PLR, of different connections can be maintained.

A large number of telecommunications manufactures have announced that they will develop ATM switches. However, it may take another one or two years before the ATM switch products can be seen in the market. For those available ATM switches, the switching architectures, output contention resolution schemes, and technologies are addressed.

2.3.1 Switching Systems

2.3.1.1 Fujitsu's FETEX-150 ATM/STM Switch

The switch architecture is named MultiStage Self-Routing (MSSR) switch [2-18]. To increase the switch throughput, the switch speed is faster than the link speed. The internal blocking and output contention problems are resolved using the 3-stage configuration, buffered self-routing modules (SRMs), and the token reservation scheme. The 3-stage configuration creates multiple paths between each input and output pair. The path selection is done at the call setup phase by the call processing module.

The self-routing module basically is a buffered crossbar. A new arrival packet is stored in the buffer at the crosspoint between the inlet and the destined outlet. To reduce the buffer size, all the buffers belonging to the same outlet are shared by different crosspoints. Since multiple packets stored at different buffered crosspoints may be destined to the same outlet, a token ring is established to resolve output contention for each outlet.

The technology of the large scale integrated chips (LSICs) is Bi-CMOS logic gate array with ECL interface.

The ATM switch is sold as a whole package. They do not sell individual switching chips.

2.3.1.2 AT&T BNS-1000 Fast Packet Switch

The BNS is a cell relay switch based on ATM protocol. This switch is designed to be used with switched multimegabit data service (SMDS), X.25, ISDN, frame relay, and other broadband services. The switch architecture is not available.

2.3.1.3 Adaptive ATMX Switch

The switch will offer service for local area network (LAN) traffic. The switching architecture uses a crossbar. The link speed is 100 Mbit/sec. The switch capacity is 1.2 Gbit/sec. The switch with one six-port card sells at price $45000. It is worth mentioning that the Transwitch is licensed by Adaptive to design the ATM segmentation/assembly chip sets. The ATM adapter card for SPARCstation is sold for $4500.

2.3.1.4 MPR Teltech AtmNet Switch

The switch offers services for LAN, WAN, and telephone carrier networks. The ATM switch module has a size of 4 X 4. It is a nonblocking switching fabric and the link speed is 160 Mbit/sec. The switching architecture is not available. The switch module adopts output queueing scheme. An ATM switch is comprised of switch modules, transmission cards, service adaptation cards, power supply cards, alarm cards, and controller cards. Every card contains a microprocessor. The switch supports multicast function and

bandwidth management. The switch can be expanded up to 32 x 32 by cascading switch modules in multiple stages. The price of the switch module is $3200.

An eight port switch card is under development. The switch will support 620 Mbit/sec link speed.

2.3.2 Switching Chips

2.3.2.1 Triquint (TQ8016) Multicast Crossbar

The architecture uses a centralized control approach [2-20]. The switching architecture can be considered as N N x 1 multiplexers (selectors) stacked in parallel. There are two sets of N output registers, where one output register is for one N x 1 multiplexer. The Log_2 N bit-wide output register is used to select one input line from the N input lines. This two-set architecture allows the switch to operate in a ping-pong fashion. The connection between an output and an input has to be reprogrammed each time a packet comes in. The procedure of setting up a connection between an output and an input is described as follows. The output register for one multiplexer is enabled first. And then Log_2 N bits of the input address are loaded into the output register.

The switch state is completely reconfigured by loading the output registers with N input addresses (sequentially). Although the multicast crossbar is not a self-routing architecture, as long as the switch state reconfiguration time is less than 2.72 μsec, the crossbar can be used as an ATM switching fabric. An output contention resolution module is required.

The switch size is 16 x 16. The link speed is 1.3 Gbit/sec. The data interface supports ECL and the control interface supports CMOS. The power consumption is 6.8 watts. The reconfiguration time is 0.33 μsec. Using external addressing logic, a 32 x 32 switch can be constructed using 4 16 x 16 chips. The technology uses GaAs.

2.3.2.2 AMCC (S2024) 32 x 32 Crossbar

The switch size is 32 x 32. The chip supports two link speeds: 400 Mbit/sec and 800 Mbit/sec [2-21]. For synchronous operation, a clock is required to feed into the chip. The link speed is 400 Mbit/sec. For transparent operation, no external clock is required. The link speed is 800 Mbit/sec. The data interface supports ECL and the control interface supports TTL. The power consumption is 9.9 watts. The reconfiguration time is 0.21 μsec. A 64 x 64 switch can be constructed using 4 32 x 32 chips without any external addressing logic. The technology uses Bipolar. An output contention resolution module is required.

2.3.2.3 Vitesse (VSC864) 64 x 64 Crossbar

The switch size is 64 x 64 and the link speed supported is 200 Mbit/sec [2-19]. There are two operations depending one whether an external clock is required. For clocked operation, a clock is required to feed into the chip. For flow through operation, no external clock is required. The data interface supports ECL and the control interface supports ECL.

The power consumption is 10 watts. The reconfiguration time is 0.44 μsec. A 128 x 128 switch can be constructed using 4 64 x 64 chips with external addressing logic. The technology uses GaAs. An output contention resolution module is required.

A general comparison (in terms of size, port speed, power, reconfiguration time, skew time, price, etc) among the three commercially available crossbars is provided in Table 2-5.

Table 2-5: A General Comparison Among Three Commercially Available Crossbar Switching Chips

Manufacturer	size	port speed	interface	broadcast mode	power	reconfig. time
Triquint (TQ8016)	16	1.3 Gbit/s	data ECL ctl CMOS	D0 to O0-O15	6.8 watts	0.33 μs
AMCC(S2024)	32	400 Mbit/s 800 Mbit/s	data ECL ctl TTL	no	9.9 watts	0.21 μs
Vitesse(VSC864)	64	200 Mbit/s	data ECL ctl ECL	no	10 watts	0.44 μs

lower speed operation	expandability	diagnostic	availability	price
yes	4 chips for expansion to size 32 and ext. addressing logic.	no	now	$223.0
yes	4 chips for expansion to size 64.	no	now	$700.0
yes	4 chips for expansion to size 128 and ext. addressing logic.	examine the contents of control register and its operation	now	$1096.0

space radiation	modes	Time Skew	crosstalk	heat sink
GA Implementation	transparent	0.4 ns	not available	thermalloy
Bipolar Implementation	synchronous transparent	0.5 ns 0.5 ns	not available	come with the chip
GA implementation	clocked flow through	1.5 ns 2.4 ns	insignificant for digital	(2.3 inch) IERC E079X2.30B

fault tolerance	reliability	number of pins	BER	propagation delay
no	not available	132	not available	1.2 ns
no	not available	196	10^{-12}	2.96 ns
no	50 FITS	344	10^{-13}	6.5 ns

2.3.3 Experimental Switching Systems/Chips

2.3.3.1 Alcatel 16 x 16 ATM Switching Element

The link speed is 600 Mbit/sec and the size of the switching element is 16 x 16 [2-22]. A shared buffer is provided in the switching element. The switching element provides multicast function and priority control. Since the switching elements are buffered, contention resolution uses the store-and-forward approach.

The technology forecast in 1992 is the CMOS.

2.3.3.2 Hitachi 32 x 32 Shared-Buffer ATM Switch

The link speed is 155.52 Mbit/sec and the switch size is 33 x 33, where 32 ports are used for data and one port is used for control [2-23]. The switch uses a shared buffered memory switch (SBMS) architecture. The switch has a shared buffer of size 4096 cells. The buffer size is reduced compared with that of non-shared buffer switching architectures. Since the switch uses the internal buffering approach, store-and-forward is used for contention resolution. A 32 x 32 155.52 Mbit/sec switch can be converted into an 8 x 8 600 Mbit/sec switch by modifying the control logic. The switch provides multicast function and

priority control. The 32 x 32 switch can be expanded to a large-scale switch. The technology used is CMOS.

2.3.3.3 Toshiba 8 x 8 Shared-Buffered ATM Switch

The link speed is 155 Mbit/sec. The switch has a shared buffer of size 184 cells. Contention resolution uses the store-and-forward approach. The 8 x 8 switch can be expanded to a large-scale switch using the Clos switching architecture. To increase the switch throughput, flow control between switching elements is employed and the switch speed is faster than the link speed. Multiplexing sequence for input lines is rotated to achieve the fairness of sharing the buffer. The technology used is Bi-CMOS.

A complete 64 x 64 ATM experimental switching system has been developed using the 8 x 8 switching elements [2-24]. They claim that the switch system exhibits the best switching performance (PLR and delay) comparing with other reported switching systems.

2.3.3.4 Mitsubishi ATM Switch

The switch operating frequency is 78 MHz. The technology uses Bi-CMOS. The switching architecture is not available.

2.3.3.5 NEC ATOM Switch

A trial ATM switching system has been built at NEC [2-25]. The architecture of the ATM output buffer modular (ATOM) switch uses a time division multiplexing (TDM) bus with output buffering. This is a contention-free switching architecture. All the input streams are multiplexed into a high-speed TDM stream. An address filer at each output port is used to select the cell destined to itself. The basic switch module has a size of 8 X 8. The capacity of the switch is 2.5 Gbit/sec. Multicast function and priority control are supported. When the input loading is less than 0.9, the PLR is less than 10^{-9}. Traffic monitoring device at the output port is also provided. A large switching architecture is constructed using the Clos topology. The technology uses is CMOS.

2.3.3.6 BellCore Sunshine Switch

The Sunshine switch uses the combination of the sorting network and the banyan network to achieve the internal nonblocking condition [2-26]. To reduce output contention, multiple (k) banyan networks are stacked in parallel to provide multiple paths. An output queue can receive up to k packets at a time. If the number of packets destined to an output port is larger than k, the overflow packets will be recycled back to an shared queue for future reentry. The switch combines the output queueing scheme and the shared recirculating queueing scheme. The switch supports priority. The technology used is CMOS.

The main components of the Sunshine switch are the Batcher-Banyan chip set which contains a 32 x 32 banyan chip and a 32 x 32 batcher chip. The Batcher and banyan chips run at bit rates of 170 Mbit/sec.

Based on the above chip set, BellCore has built another 32 x 32 fast packet switch prototype that adopts input buffering scheme. The input ring reservation scheme is used to resolve the output contention. BellCore has also used the 32 x 32 chip set as a basic module to build a 256 x 256 sorted-banyan-based switch with a total capacity of 35 Gbit/sec.

2.3.3.7 NTT 8 x 8 Cross-Connect

The switch architecture adopts input queueing [2-27]. The switching fabric uses the Batcher-banyan point-to-point nonblocking switching architecture.

The output contention resolution scheme uses a combination of time scheduling, pipeline processing, and input ring reservation scheme with a large checking depth. The main advantages of this approach are that the operation of the reservation speed is independent of the switch size since the output reservation is performed in a pipeline fashion.

The packets (at the input queues) which have successfully reserved the output ports in advance will be transferred to another queue, called sending queue. In the sending queue, each buffer space corresponds to one future slot.

2.3.3.8 Siemens ATM Switch

The switching element size is 16 x 8. A shared buffer architecture is used for each switching element [2-28]. All the packets are read into the memory first. The packets are read out from the memory to different output ports for transmission. To reduce the memory access time, a wide parallel bus is used for memory interface. The technology used is CMOS. A 32 x 32 ATM switching module is constructed using 12 switching elements. Priority control is provided. Each input port uses the leaky bucket scheme to monitor and policy the incoming traffic.

2.3.3.9 AT&T Bell Laboratories Switches

AT&T Bell Laboratories has researched several fast packet switches. The first one is the Starlite packet switch, which consists of a Batcher's sorting network, a Trap network and a banyan network. The banyan network is an unbuffered switch, and because of the sorting network, there is no internal blocking. The output contention problem is resolved by the Trap network, which circulates the duplicate address packets to the reentry input ports. The second fast packet switch is a buffered banyan network. There are two buffers for each input within one switching element and the buffer size is one packet. The internal blocking and output contention problems are tackled using the internal buffering approach. The switch speed is higher than the link speed to further reduce the blocking problems within the switch.

2.3.3.10 IBM Switch

IBM has developed a fast packet switch prototype. The switch supports a 45 Mbit/sec line speed and optical interface. Routing is based on the source routing method, where the packet carries end-to-end path information. The switching architecture is not available.

2.3.3.11 Fujitsu HEMT Switch

A high electron mobility transistor (HEMT) ATM switch LSI has been developed by Fujitsu and its Laboratory. A prototype unit is constructed using the LSI module.

2.3.4 Future Plans

2.3.4.1 BBN Emerald Switch

The switch uses the off-the-shell components. The switch size is 10 x 10. The link speed is 160 Mbit/sec. The switch uses busless design and distributed processing. It is expected the design will be very similar to that of the Butterfly supercomputer, where the architecture adopts the buffered banyan approach. The switch will be available in Aug. 1993.

2.3.4.2 TRW Switch

TRW announced that two ATM switches will be built. The first model, 2001, has capacity of 720 Mbit/sec. The other model, 2010, has a capacity of 2.88 Gbit/sec. The switching fabric is nonblocking. Two sizes are supported. The first one has a size of 64 x 64 and the link speed is 45 Mbit/sec. The second one has a size of 16 x 16 and the link speed is 155.52 Mbit/sec.

A prototype is under development. The products are coming out at the third quarter of 1993. The price of the whole switch is at the range of 1 million dollars.

2.3.4.3 University Optical Switch

Columbia university builds an experimental optical switch. The switching architecture uses wave-length division multiplexing. The capacity of the switch is 1 Gbit/sec.

2.3.4.4 GTE Government System Switch

The switch provides services for LAN and WAN. The switch called secured prioritized ATM node (SAPNode) has link speed of 155 Mbit/sec. The switch size is 64 x 64.

2.3.4.5 Cabletron System Switch

The ATM will be in production at the third quarter of 1993. The switch is provided as backbone router to send packets among different types of LANs.

2.3.4.6 Synoptics Communications Switch

A local ATM switch will be in the market next year to support high-speed desktop applications.

2.3.4.7 NEC Switch

NEC plans to provide an ATM switch to compete with AT&T and Fujitsu Network Systems. The switching system will provide interfaces for cells, frame relay, and synchronous optical network (SONET).

2.3.4.8 DSC Switch

A fast packet switch will be available for field trial at the end of this year.

2.4 The Proposed Switching Architecture

The proposed multicast switching architecture is the multicast crossbar switch (see Figure 2-11). The multicast crossbar switch is selected for the following reasons [2-29]:

- it is commercially available
- it is the switching architecture chosen by most terrestrial switch manufactures
- the switching delay is low
- the structure is simple
- the switching fabric is point-to-multipoint nonblocking
- the operation characteristics (such as power) are very suitable for on-board applications.

The input queueing strategy is selected for easy implementation and low complexity.

The above selection matches with the recommendation made in Phase 1 report. In Phase 1 report, three multicast switches were evaluated based on power consumption, application specific integrated circuit (ASIC) count and fault tolerance. These three multicast switches are the self-routing multicast crossbar, the sorted-multicast-banyan switch, and the switch with multicast modules at the output port. The self-routing multicast crossbar was chosen as the optimal architecture because it has the lowest power consumption per port and the lowest ASIC count for the switching fabric.

Due to HOL blocking of input-queued switches, the switch throughput (for point-to-point connections) can not exceed 58% for a larger N. To increase the switch throughput, two approaches are possible. The first approach is to design a very efficient scheduling algorithm for packet transfer, such as using the centralized ring reservation scheme with a large checking depth. The other approach is to increase the switch speed. More discussion and recommendation on the output contention control scheme for a multicast crossbar switch is provided in Section 3.1.

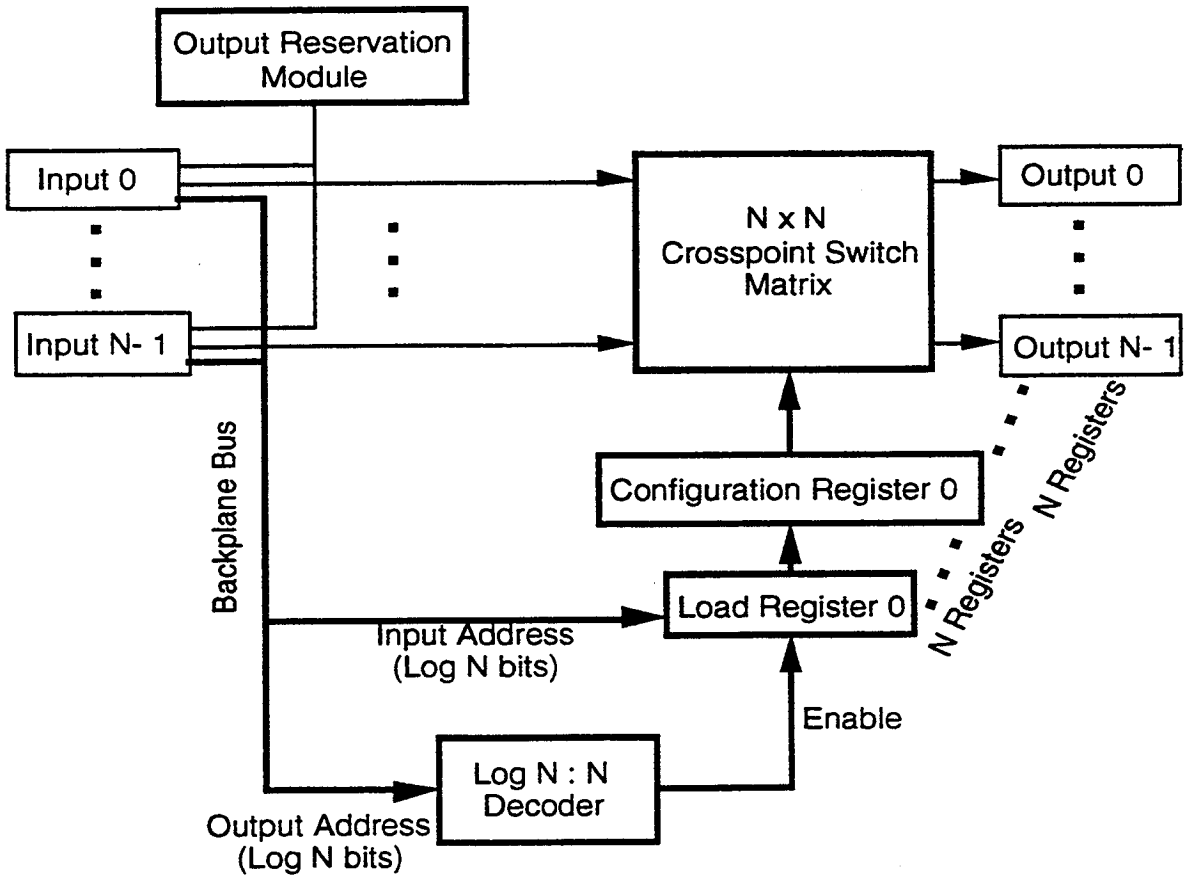


Figure 2-11: Proposed Multicast Switching Architecture: Crossbar Switch

Section 3

Design Considerations for Switching Subsystem

This section addresses design considerations for the switching subsystem based on the selected architecture (i.e., the crossbar). They include output contention resolution, satellite virtual packet format, priority control, integrated operation of circuit and packet switched traffic, and fault-tolerant design. Based on these analyses, high-level functional requirements for the on-board baseband switching subsystems are presented in Section 4.

3.1 Output Contention Resolution

There are two major system design issues associated with an FPS. The first issue, which is the subject of this subsection, is the output contention. The second issue, which is the subject of the task "Critical Element Design and Simulation", is the congestion problem. Since there are no preassigned routing paths for ATM cells in a fast packet switch (as in a circuit switch), several packets from different input ports may be destined to the same output port at the same time. This situation is referred as output contention (see Figure 3-1). If this occurs, output contention resolution has to be performed such that only one packet is allowed to be transmitted to the output port. The other packets have to be buffered or dropped depending on the switching architecture. The output contention can be resolved using a special switch structure or using a mechanism to avoid output contention. Various output contention resolution mechanisms for different fast packet switching architectures have been addressed in Reference 3-1. The discussion in this section contains a summary of Reference 3-1 for the selected switching architecture and some new output contention resolution mechanisms proposed recently.

Based on the contention level encountered in the packet switch, the switch architectures are categorized into two classes: contention-free switch and contention-based switch. A contention-free fast packet switch is a switch whose output port can receive up to N packets in one link slot time, where N is the size of the switch and a link slot is defined as packet size/link speed. Within the contention-based switch class, the switch architectures are classified according to the output contention resolution mechanism (or packet transfer scheduling algorithm). There are three approaches: the first one employs an output reservation scheme at the input ports, the second one uses a path setup strategy to resolve blocking within the switching fabric and output contention at the same time, and the third one uses an address filter at the output port. For the selected crossbar switching architecture, only the first approach is applicable. The other two approaches will not be discussed in this report.

3.1.1 Contention-Free Switches

The prerequisite of a contention-free (multicast) switch is that the switching fabric must be point-to-multipoint nonblocking. A contention-free switch is constructed by running the switch speed N times faster than the link speed or stacking N switching fabric in parallel. Clearly the hardware cost may be too high to have any practical applications when the switch size or the switch capacity is large. The contention-free switch is mainly used for a switch with a small capacity requirement.

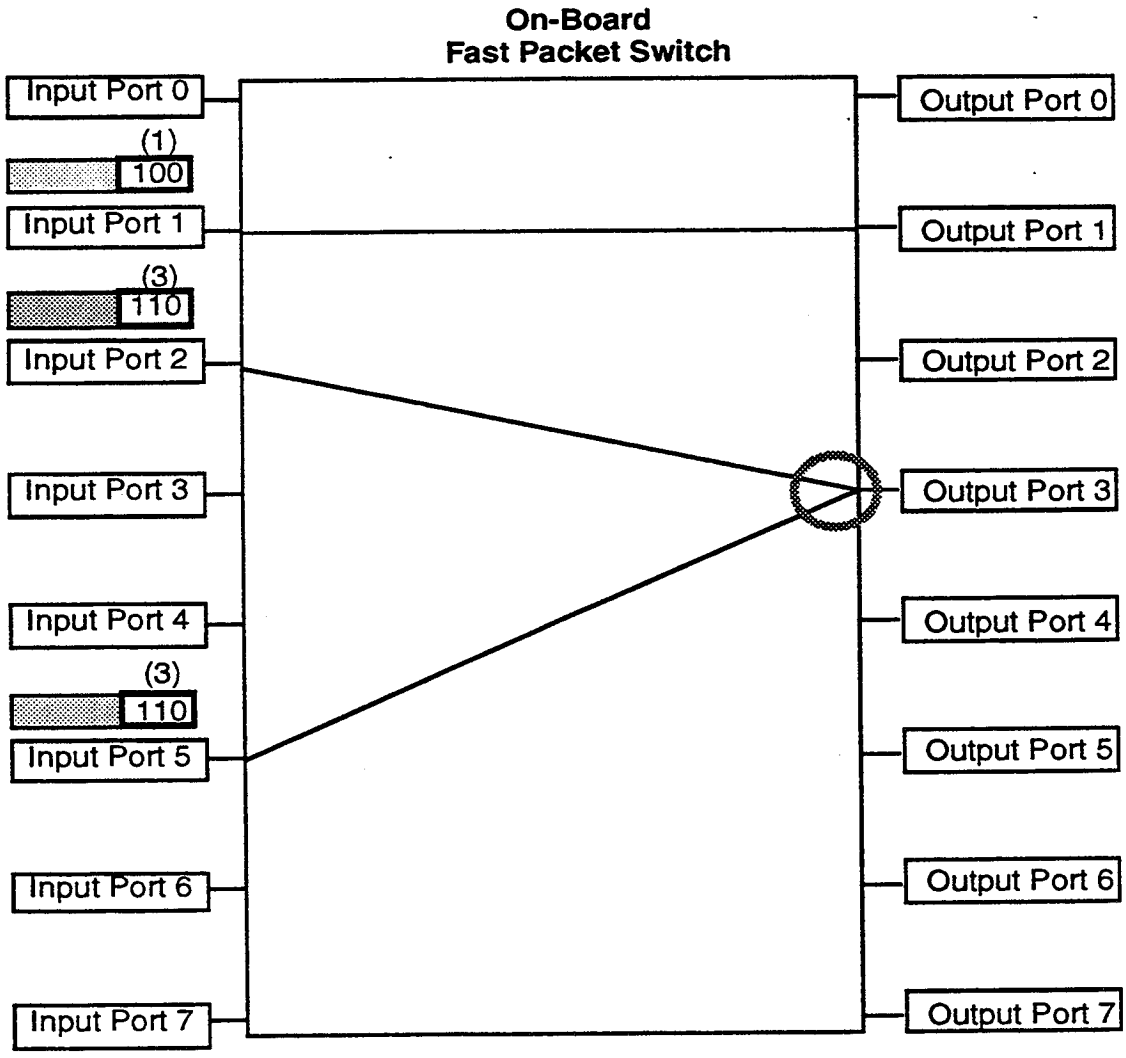


Figure 3-1: An Illustration of Output Contention in A Fast Packet Switch

3.1.2 Output Reservation Scheme for Contention-Based Switches

Allowing some output contention to occur in the switch can reduce the hardware cost and speed requirement compared with the contention-free switch. To resolve the output contention of a crossbar switch with input buffering, packet transfer at the input ports has to be scheduled. In each packet transfer process, a non-contending set of connections (or a permutation set of connections) is chosen from the packets at the input ports. The choosing criteria may be based on priority, time stamp, a specific order, queue length, or random. The packets presented to the switching fabric all have distinct destination addresses and the packets will not be collided in the switching fabric and the output ports.

Due to head of line (HOL) blocking at the input port queue, the packet switch throughput for point-to-point connections cannot exceed 58% for a large N [2-12]. The throughput is defined as the average number of packets arrived to the output ports in one link slot divided by the switch size, where a link slot is defined as (packet size/link speed). This blocking is a side effect of the results of output contention. Assume that one packet at the head of input queue cannot be transmitted due to output contention. Then this blocked packet hinders the delivery of the next packet in the queue due to the first come first serve (FCFS) nature of the queue, even though the next packet can be transmitted to the destination without any blocking. To improve the throughput of the switch, there are three basic methods. The first method is to increase the switch speed so that more than one packets in the input port can be processed within one link slot time. The ratio of the switch speed to the link speed is defined as the speedup factor (S). In one link slot time, S packets at an input port are processed by the output port reservation module. An input port is allowed to transmit from one to S packets in one link slot time. The second method is to use p parallel switches, p transmitters at the input port, and p receivers at the output port. The result is there are p disjoint paths between each input and output pair, the input can transmit up to p packets, and the output port can receive up to p packets at the same time. The third method is to design a more efficient scheduling algorithm to increase the throughput of the switch. In the first two methods, since more than one packets can arrive at one output port in one link slot time, the switch has to incorporate output queueing to hold the packets. In this case, each output port performs as a statistical multiplexer. Since output queueing is used, the throughput definition is modified as the average number of packets leaving the output ports in one link slot divided by the switch size. The third method does not need any output buffering.

The output reservation scheme can be performed centrally or in a distributed fashion. In the centralized scheme, output reservation for different input ports is executed by one common module (the output reservation module). In the distributed scheme, output reservation for different input ports is executed by different modules. The decision made by one module is independent of the others. The ring reservation scheme is chosen as the representative for discussing the centralized scheme due to its simplicity, easy implementation, and versatile applications [3-2]. The two-phase output reservation scheme is used to discussed the distributed scheme [3-3]. A new scheme, which combines the advantages of the ring reservation scheme and the two-phase reservation scheme is also introduced in this subsection.

The following nine subjects are addressed in this subsection:

- centralized ring reservation scheme for point-to-point switches
- centralized ring reservation scheme for point-to-multipoint switches
- centralized ring reservation scheme for link grouping
- centralized ring reservation scheme for parallel switches
- centralized ring reservation scheme with pipeline implementation
- application of centralized ring reservation scheme to crossbar switch
- decentralized reservation scheme
- a new reservation scheme
- proposed reservation schemes for a crossbar switch

The priority control in conjunction with the centralized ring reservation scheme is discussed in Section 3.3.

3.1.2.1 Centralized Ring Reservation Scheme for Point-to-Point Switch

Basically, the ring reservation scheme uses the token ring principle to resolve output contention [3-2]. The input ring connects all the input ports of the switch and the function of the ring is to perform output reservation for each input port. At the beginning of every slot time, the output reservation module sends a stream of tokens and passes these tokens through all the input ports, where one token represents one output port (see Figure 3-2). It is assumed that the tokens are passed serially from one input port to another input port. The input port searches the right token according to the destination routing tag of the current head of line (HOL) packet. If the token for the corresponding routing tag is on the stream, then the token is removed so that no other input ports can transmit a packet to the same output port at the same slot time. After the token stream has circulated through all the input ports, the input ports (that have reserved a token) can transmit the packet at the beginning of the next slot time. In implementation, only one bit is necessary for one token. For example, value 1 represents there is a token and value 0 no token. In the example shown in Figure 3-2, the token streams start from input port 0. The HOL packet buffered at Input port 0 is destined to output port 1. Consequently, input port 0 takes token 1, i.e., changes the bit at position one of the output map from 1 to 0. The HOL packet buffered at input port 1 is also destined to output port 1. Since token 1 has been taken (by input port 0), the HOL packet at input port 1 is blocked. To assure fairness among the input ports of accessing the tokens, several ways can be employed. The first is at different slot time, the stream will be started at different input port. The second is to send this stream from the beginning of the input ports and from the end the input ports alternatively.

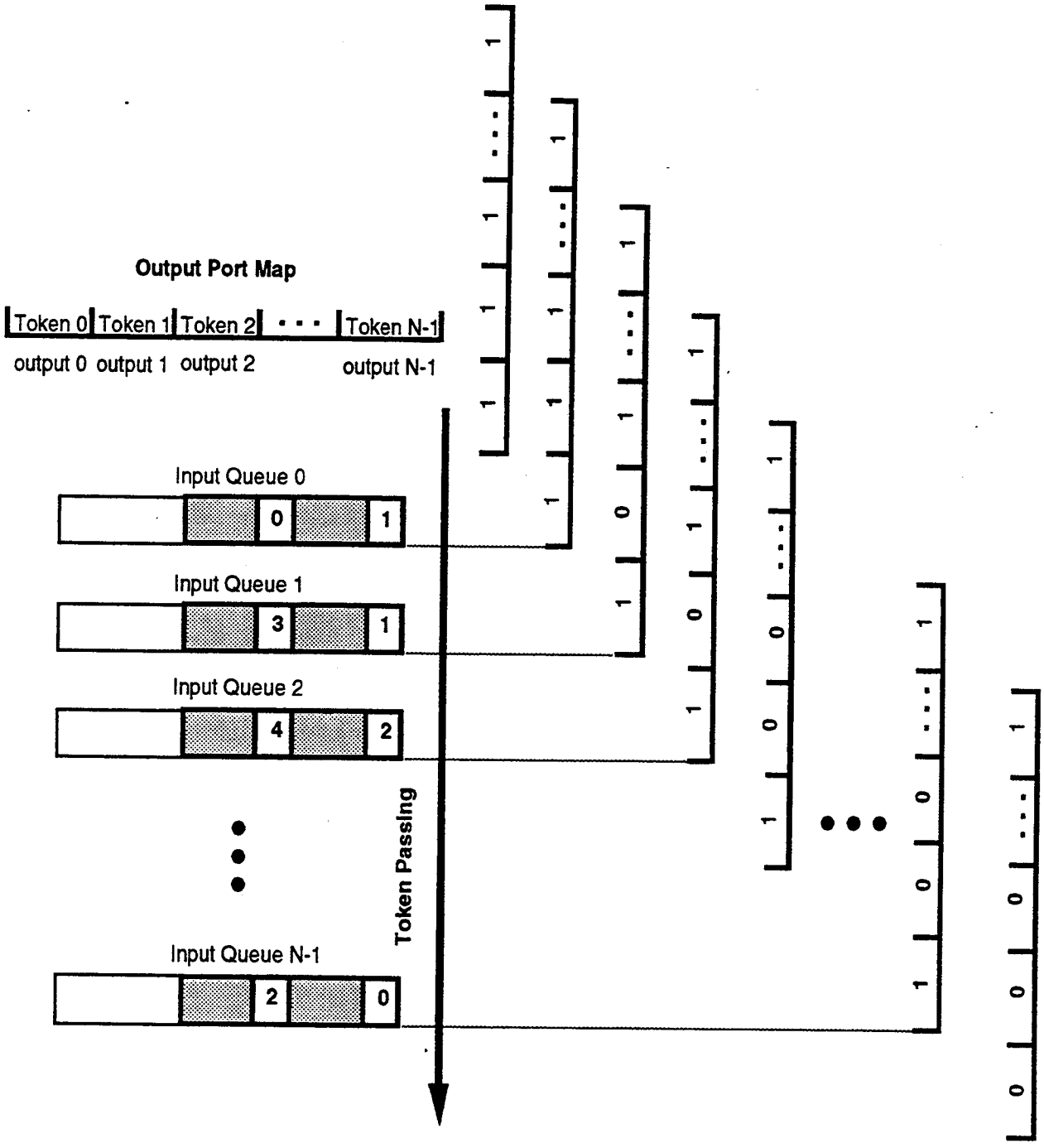


Figure 3-2: An Illustration of Input Ring Reservation Scheme

As previously discussed, the throughput of the input-buffered fast packet switch is limited due to HOL blocking. One way of improving the throughput is to use a non-FIFO input queue. If the first packet is blocked due to output contention, the input port also checks the packets at the back of the first packet in the queue. This scheme is also referred to input queue by-pass. The number of packets examined each time depends on

the preset window size or the "checking depth". If one of the packets within the checking depth has a chance to be transmitted, this packet will be transmitted first. In this sense, a FIFO input queue has a checking depth 1 while a non-FIFO input queue has a checking depth greater than 1. Theoretically, if the checking depth is infinite, the throughput of the switch can reach 1. From the simulation results [2-11] [3-1], the improvement of the switch throughput decreases when the checking depth gets larger. Please refer to Table 2-2 for switch throughput for different checking depths. Hence, in practical, the checking depth is less than O (10). To further improve the throughput, either increasing the switch speed or using parallel switches is necessary.

3.1.2.2 Centralized Input Ring Reservation Scheme for Multicast Switch

The input ring reservation scheme used in the point-to-point switch can be directly applied to the point-to-multipoint switching fabric. The difference is that in a multicast switch the input port can reserve more than one output ports at a time, i.e., the input port can take more than one tokens at a time. If call splitting is allowed, the transfer of the multicast packet to different destinations can be partially completed. The call splitting concept is introduced in Section 2.1. If call splitting is not allowed (one-shot operation), the transfer of the multicast packet to different destinations has to be completed at the same slot time.

3.1.2.3 Centralized Input Ring Reservation Scheme for Link Grouping

The switch size does not have to be consistent with the number of uplinks or number of downlinks. If the number of uplinks is small and the link speed is very high (e.g. 600 Mbit/s), then it is not easy to implement such a high-speed switch using low-power devices on-board. One way of resolving this issue is to apply link grouping, i.e., the high-speed input links can be demultiplexed first into several lower-speed intermediate links, and these links are fed into different input ports of a switch (see Figure 3-3). Assume the total number of high-speed uplinks is g and the switch speed is S. If we use 1:m demultiplexer to reduce the link speed, then the size of switch is enlarged to g*m. The speed of the switch is decreased from S to S/m. Each packet header carries the destination address (the physical address). The physical address is to specify the downlink beam. This physical address must be translated into logical address. The logical address is used to set up a path within the switch.

Another advantage of performing link grouping is that the output contention problem of the switch can be reduced. Suppose m = 3 and consider downlink 1. Whether a packet is routed to output port 0, output port 1, or output port 2, it will be multiplexed to the downlink 1. Hence, by performing address translation at the input port and a careful design of the output port contention resolution scheme, the output link efficiency can be largely increased by reducing output port contention. The input ring reservation scheme can be directly applied in this case. There are l*m tokens in the token stream, where every m tokens is grouped into a super token for each downlink. All the packets destined to the same downlink can remove any token (any output port) in the super token (the downlink). The token removed by the input port will be the routing tag for the packet at the next time slot.

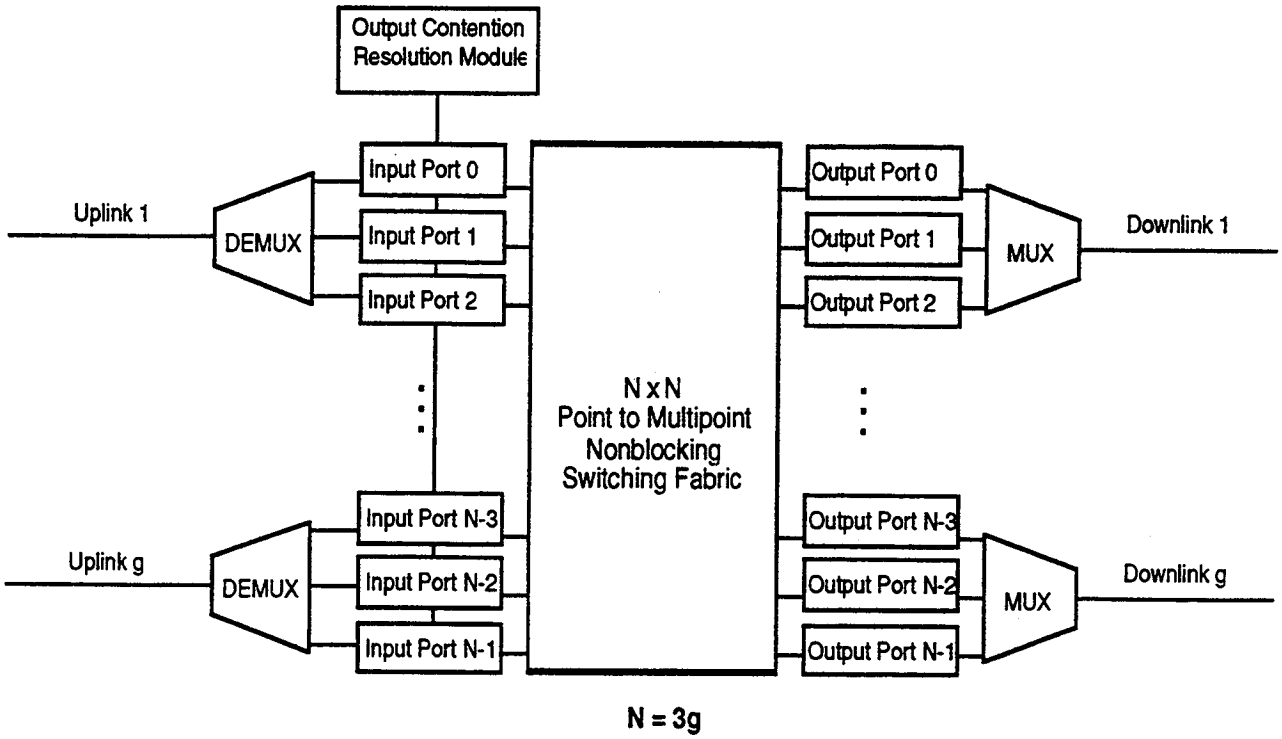


Figure 3-3: An Illustration of Link Grouping Concept

3.1.2.4 Centralized Input Ring Reservation Scheme for Parallel Switches

The input ring reservation scheme can be applied parallel switches with a simple modification. Assume there are p parallel switching fabrics, p parallel transmitters at the input port, and p parallel receivers at the output port. Since the output port has the capability of receiving p packets at the same time, the token format has to be modified. There are N*p tokens in the token stream, where p tokens are grouped into a super token for each output port. All the packets destined to the same output port can remove any token in the super token. The only restriction is that the packets at different transmitters at the same input port need to have different destinations so that out-of-sequence will not occur. With this configuration, the first TX can process the token first, and followed by the second TX, and so on. Since there are p receivers at the output port, output buffering is necessary to handle the situation that more than one packets come to the output port at the same time. The output port first multiplex the packets in the p receivers into one high-speed TDM bus and feed this TDM stream into a common buffer.

3.1.2.5 Centralized Input Ring Reservation Scheme with Pipeline Implementation

The time reservation algorithm, proposed in [3-3], uses future time scheduling, pipeline processing, and a large checking depth. The main advantages are that the reservation

speed is independent of the switch size and a large checking depth is used to improve the throughput.

Basically, this algorithm implements the tokens in parallel. The output port reservation module sends out streams of tokens in parallel. If the switch size is N, the number of parallel streams passing around the input ports is also N, where one parallel stream is for one time slot. Each input port processes one parallel stream at one slot. An example is given below to illustrate the operation.

First the output port reservation module generates a parallel stream for time slot t and sends the stream to the input port 0. All the packets within a checking depth d at input port 0 have a chance to reserve the outputs for time slot t. After this, the output port reservation module generates another parallel stream for time slot t+1 and sends the stream to the input port 0. All the packets within a checking depth d all have a chance to reserve the outputs for time slot t+1. At the same time, the parallel stream for time slot t is shifted from input port 0 to input port 1. All the packets within a checking depth d all have a chance to reserve the outputs for time slot t. The shift cycle is one slot time. The total number of shifts required for one parallel stream to shift from the reservation module to output port N-1 is N. Note in this configuration, input port N-1 has the least probability of obtaining a token since it is the last stop of the parallel token streams. For fair access to the tokens, the parallel stream has to alternate the starting input port every M slots, where $M \geq N$.

There are two ways of checking a large depth: serial search or parallel search. The advantage of parallel search is that the search time is independent of the checking depth.

Since output port reservation is performed in pipeline, packets will be scheduled into the future. The packets which have successfully reserved the output ports in advance will be transferred to another queue, called send queue. In the send queue, each buffer space corresponds to one future slot. When the future slot comes, the packet corresponding to the future slot in the send queue is sent out. The conventional scheme can be considered to have a send queue of size 1. And the future slot for the send queue is always the next slot.

There is a fixed queueing delay associated with this scheme because of pipeline operation regardless of traffic loading. An example is illustrated in Figure 3-4. Assume current slot is t. At the beginning of operation, the token streams assigned to (t+1) slot must be at input port N-1; otherwise, there is no pipeline operation. Consequently, the token streams assigned to (t+N) slot is at input port 0. The packets at input port 0 suffer a fixed queueing delay of N slots and the packets at input port N-1 suffers a fixed queueing delay of 1 slot. On average, a packet (at any input port) suffers a fixed delay of $\frac{N}{2}$ slots. In the conventional scheme, the fixed queueing delay is always one slot time.

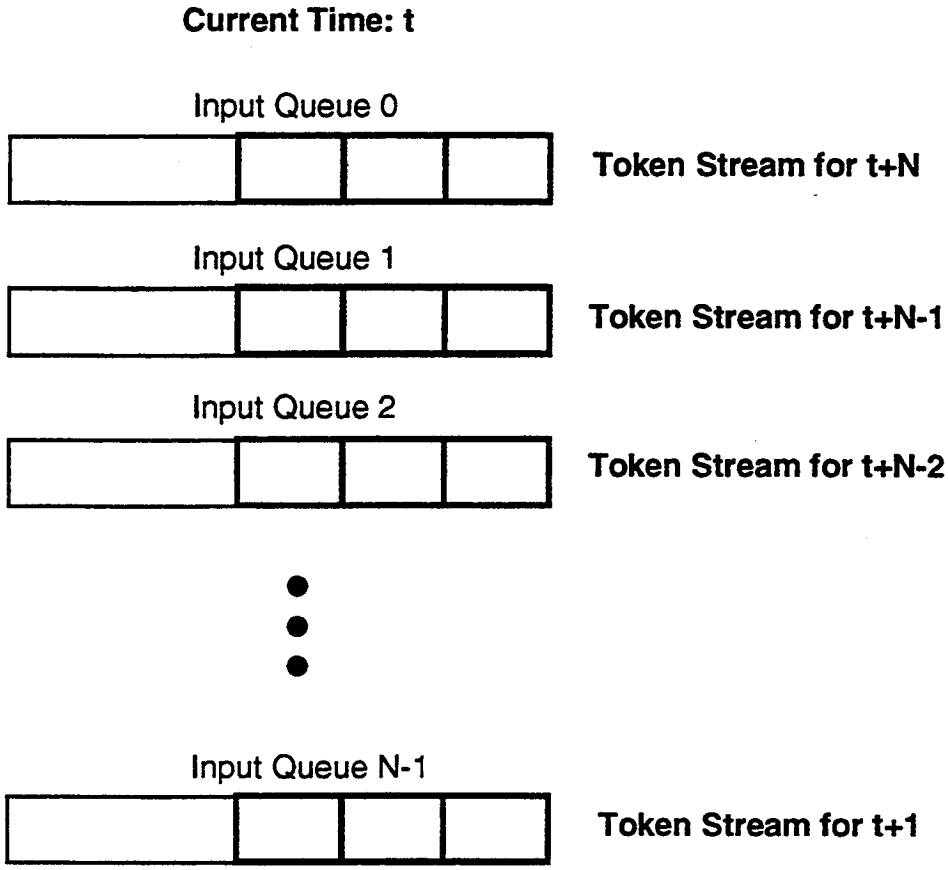


Figure 3-4: An Illustration of a Fixed Delay Introduced by The Pipeline Operation

Since the reservation is performed in a pipeline fashion, the reservation speed is independent of the switch size. The searching time into the input port to find a packet to match the arrival tokens is independent of the checking depth if search is also performed in parallel. A large check depth is feasible in this approach.

3.1.2.6 Applicability of the Centralized Input Ring Reservation Scheme to Crossbar Switches

As discussed in Section 2, high-speed crossbar switches are commercially available. Strictly speaking, these crossbar switches are not self-routing switches since crosspoint configuration for each output port has to be performed sequentially not in parallel. Nevertheless, the self-routing tag of each packet can be stripped off and used to select (enable) the output control register, and at the same time the input address is loaded into the register. After this, the input ports whose packet has reserved an output port can start transmission at the beginning of the next slot time. This process, shown in Figure 3-5, has to be executed for each input port sequentially. As long as the switching states of the

switch can be reconfigured in one packet time, the packet transmission through the switch is not disrupted.

The sequential loading of the input port address into the register makes the centralized ring reservation scheme even more attractive. Remember that the tokens represent the destinations which are granted for transmission at the next slot time. The tokens in conjunction with the input addresses can be used for switching state configuration. There are N bits for N destinations in the token stream. There are N load registers with Log_2 N bit wide in the crossbar switch, one for each output. Two approaches are identified to reconfigure the states of the switch.

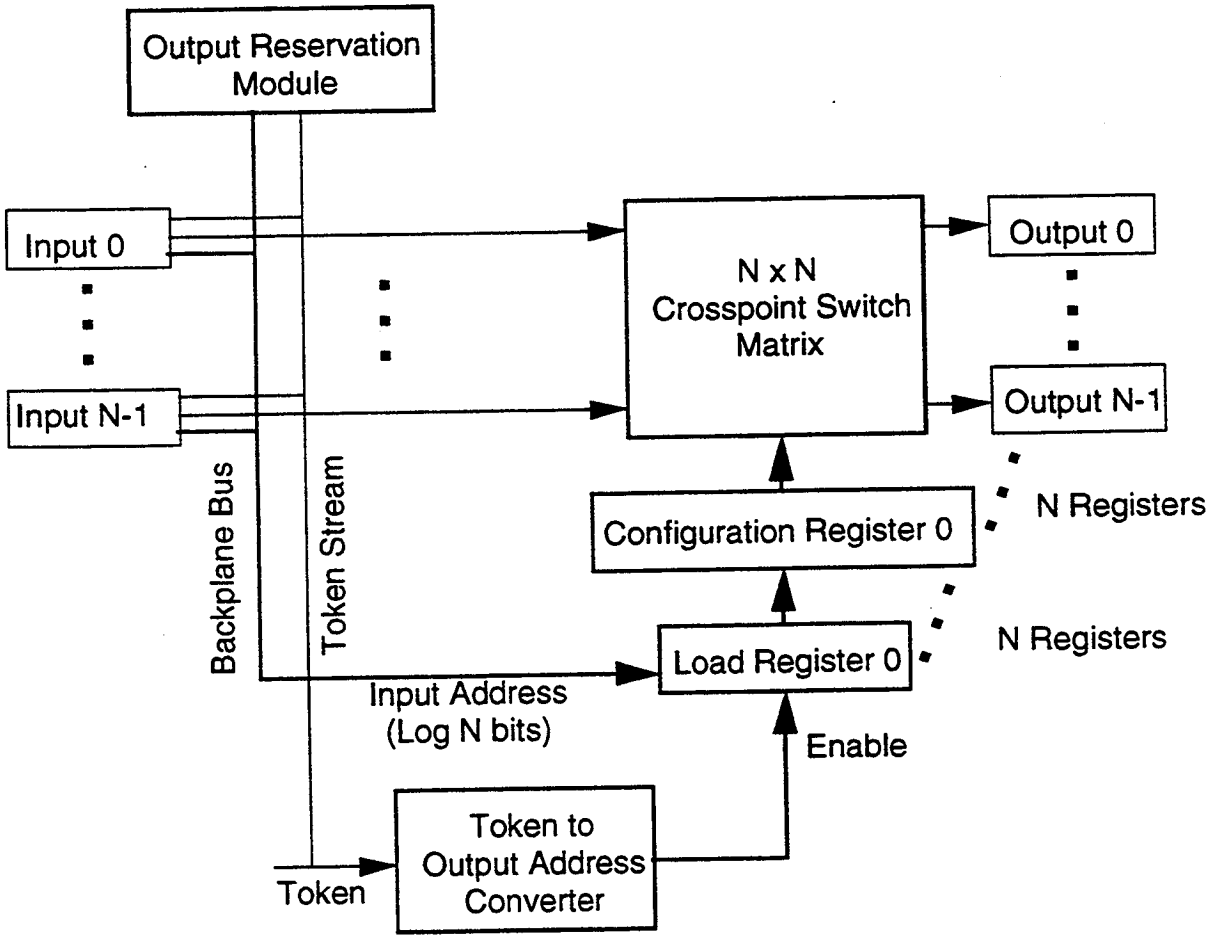


Figure 3-5: High-Level Design of Applying Input Ring Reservation Scheme to Crossbar Switch

The first scheme is to let the input port load its own address into the load register following a specific sequence. The specific sequence is based on the destination address of the packet waiting for transmission at the input port. Each bit in the token stream is used to enable the corresponding load register. For example, the first bit in the token stream is used to enable load register 0, the second bit in the token stream is used to enable load register 1, and so on. Each input port scans the output port address of the packet (which has reserved an output port) waiting for transmission. If the output port address is 0, the

input address (Log_2 N bits) of the input port is loaded into the register after loading register 0 is enabled. After this, load register 1 is enabled. The input port, whose packet waiting for transmission is destined to output 1, loads its own address to register 1; and so on.

The second scheme is to append the input port address after the token stream once an output port is successfully reserved by the input port (shown in Figure 3-6). For example, assume the packet at input port 2 reserves output port 0. Then address 2 will be inserted in I_0 position. If the input port fails to reserve an output port, the input port address is not appended. The input port addresses will be loaded into a centralized controller. The centralized controller configure the states of the switch sequentially using the input port addresses. After the switching state has been configured, the input ports, whose packet has reserved an output port, can start transmission.

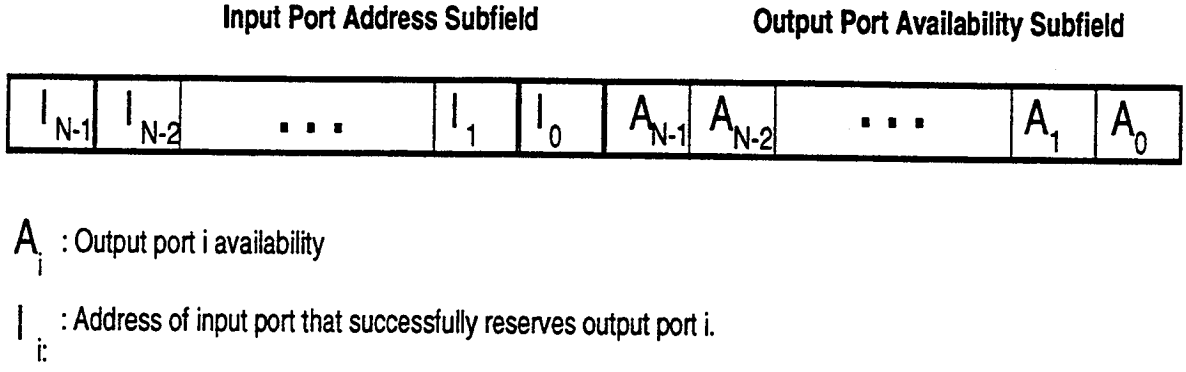


Figure 3-6: Token Format with Input Port Subfield

3.1.2.7 Decentralized Reservation Scheme

The distributed reservation scheme has an output reservation module for each output port [3-3][3-5]; however, these reservation modules do not have to be in separate physical entities. The scheme is explained in detail below.

As illustrated in Figure 3-7, there are N reservation modules for N output ports. In the output port reservation process, each input port has two phases: request and arbitration. In the request phase, each input port sends a request to reservation module i, where i is the destination address of the HOL packet in the queue. Each reservation module may receive up to N requests from N different input ports at one time. After the module receives the requests, the reservation enters the arbitration phase. There are two different designs depending on whether future scheduling is allowed.

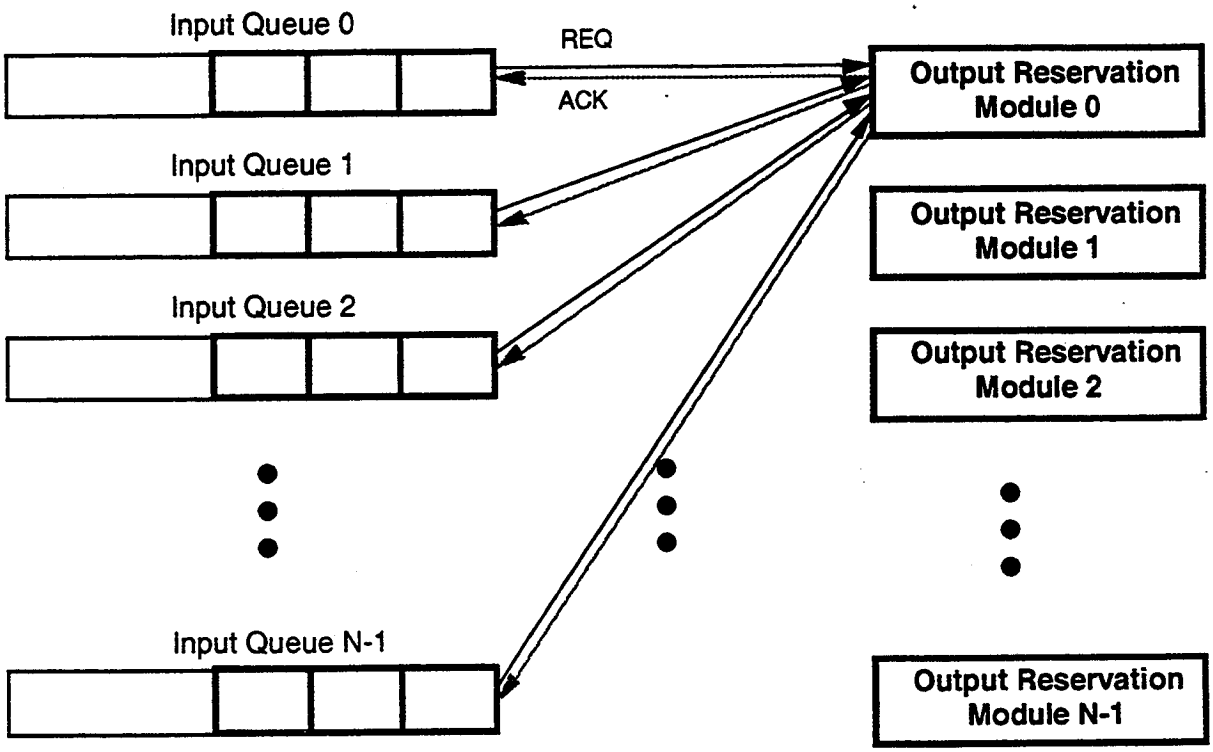


Figure 3-7: The Configuration of Decentralized Reservation Scheme

Case 1: No future scheduling is allowed. In this case, each module schedules the requests for the next transmission slot. If the module only receives one request, an ACK is sent back to the input port. The input port which receives an ACK can send out the HOL packet at the next slot. If the module receives more than one request, the module selects one of the requests randomly. The input port whose request is selected is notified by an ACK. The input port whose request is not selected will not be notified. Due to HOL blocking, the saturation throughput is less than 58% for a large N.

Case 2: Future scheduling is allowed. Each reservation module keeps a variable, the next available transmission slot, in memory. When the module receives multiple requests, every request is assigned a future, nonconflicting time slot. The future transmission slot assignment is sent back to each input port. Since each output reservation module makes its own transmission time assignment, conflict of assignments from different modules may occur at an input port. Assignment conflict occurs when two or more packets in the queue are assigned the same transmission slot. The input port must arbitrate the assignments. An example is given below to illustrate the assignment conflict. Assume each of the three input ports (1, 3 and 4) sends a request to module 0. The next available transmission slot for module 0 is t_0. Module 0 assigns t_0 to input port 1, $t_0 + 1$ to input port 3, and $t_0 + 2$ to input port 4. The next available transmission slot for output port 0 becomes $t_0 + 3$. The update of the variable for the next transmission slot can use a counter.

At the next slot time, assume each of the two input ports (0 and 3) sends a request to module 1. The next available transmission slot for module 1 is t_0. Module 1 assigns t_0 to

input port 0 and $t_0 + 1$ to input port 3. The next available transmission slot for output port 1 becomes $t_0 + 2$.

When an input port receives an assignment from a module, it assigns the transmission slot to the corresponding packet in the queue. Input port 3 receives two assignments (for two packets) with the same transmission time ($t_0 + 1$). This represents an assignment conflict. If this occurs, the input port discards all but one of the conflicting assignments. The packets whose assignments are discarded enters the request phase again. In this case, input port 3 discards the one of the two assignments, say the assignment from module 1. Input port 3 sends another request to module 1 again. The transmission slot for each packet in the input queue is stored in a control memory. The control memory reads out the packet when the system time matches with the transmission slot time. Since the slots can be scheduled into the future, the saturation throughput can reach 62% [3-5].

In the above scheme, when a transmission slot is conflicted with the previous slots at an input port, the conflicted transmission slot is discarded. The wasted transmission slots degrade the switch throughput. To overcome this inefficiency, the conflicting slots are recycled back to the reservation modules. These recycled slots are stored in memory. Using the above example to illustrate the recycle concept. Input port 3 sends $t_0 + 1$ slot back to the module 1. There is a limit for the maximum number of recycled slots which can be stored in the module. Using the recycle mechanism, a slot can be reassigned for infinite times until the slot is expired. If a slot can not be assigned to any input port, eventually the slot will be expired. If a slot is expired, the slot is erased from the memory. By recycling the conflicted slot back to the module, the switch throughput is increased from 62% to 92%. This throughput is achieved by allowing only one recycled slot to be stored in the module. It was shown in Reference 3-5 that no significant improvement can be achieved by allowing more recycled slot to be stored in the module compared with one recycled slot. Since only one recycled slot is allowed to be stored in memory, when a new recycled slot comes, the old recycled slot is overwritten.

Since the recycled slots are always earlier than the next transmission slot, the module should always assign the recycled slot to the request if there is any recycled slot. If there is no recycled slot, the next transmission slot is assigned. To eliminate the situation that the recycled slot is always assigned to the same input port, a simple rule is enforced. The recycled slot can not be used until an assignment (using the next transmission slot) is made. Therefore, an input port can not receive the same assignment for two successive slots. However, this rule introduces out-of-sequence problem. An example is given below. Assume input port 0 send a request to module 1. The next transmission slot for module 1 is t_1 and the module has a recycled slot t_0 ($t_0 < t_1$). Module 1 can not use t_0 until t_1 is assigned. In this case, input port 0 receives the transmission time t_1. If input port 0 sends another request to module 1, the input port 0 receives another transmission time t0. Since $t_0 < t_1$, the packets will be transmitted out of sequence. A better approach is to use the recycled slot first and dynamically update the assignment sequence for the arrival requests in a module. For example, the sequence for a module to assign the time slot for each request is based on the input port address. The first sequence is 0, 1, 2, .., N-1. The following sequences rotate one element at a time. Evidently, this approach does not work if there is only one request at the module. By using this approach, the probability that an input port receives the same assignment multiple times is reduced.

For the multicast operation, the input port sends out multiple requests to different modules. Call splitting capability for multicast operation is a necessity in this architecture. The link grouping concept can be directly applied to this scheme. One reservation module represents m output ports. The reservation module can assign any output port to the arrival requests. The decentralized scheme can be applied to a crossbar switch. Each input port uses the routing tag of the packet to enable the load register. At the same time, the input port sends out the input address to be loaded into the load register.

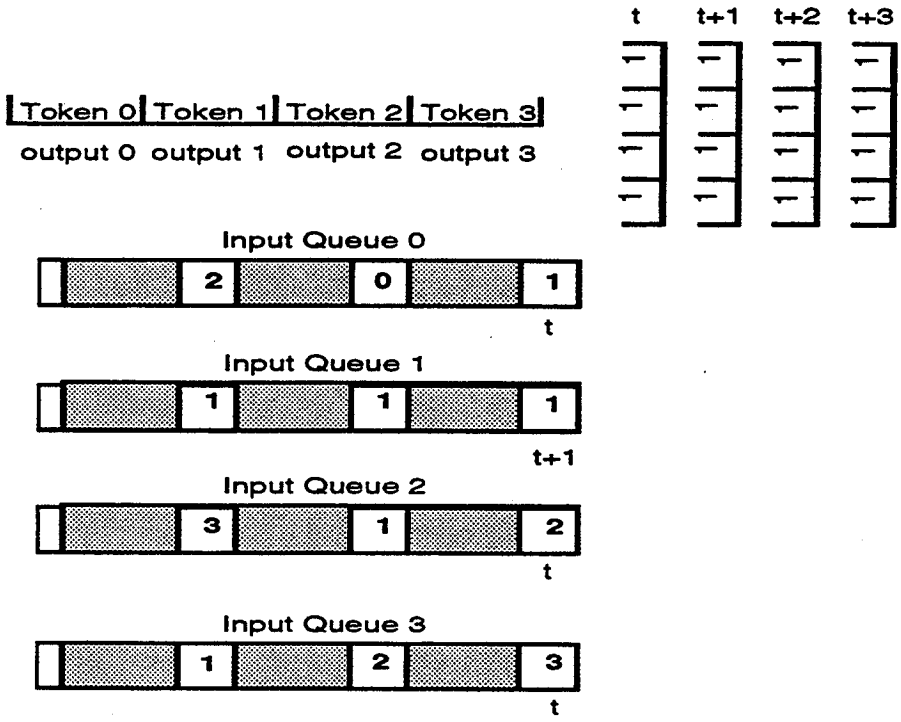
The advantage of this scheme is that no switch speed up is required to achieve a high throughput; consequently, no output queueing is required. In addition, a distributed scheme can be made more robust than a centralized scheme.

3.1.2.8 Centralized Ring Reservation Scheme with Future Scheduling

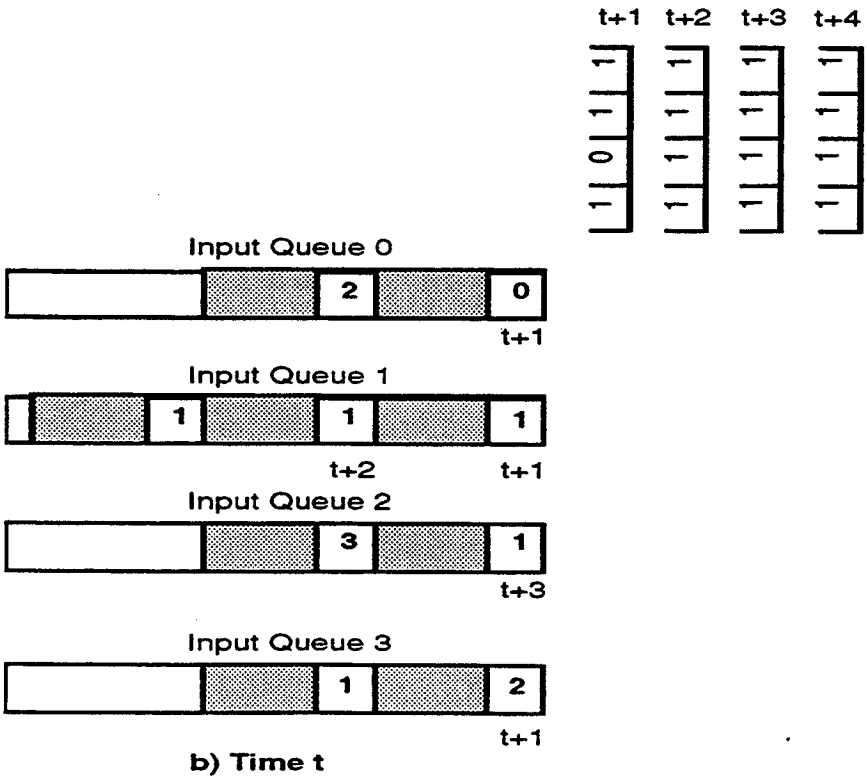
The future scheduling concept introduced in Reference 3-5 inspires a new output port reservation scheme. This scheme combines the basic centralized ring reservation scheme and the future scheduling concept. The output reservation module sends out token streams in serial. However, the number of token streams is more than one, say m. Assume the current slot time is t-1. Then the reservation module sends out m token streams for time t, t+1,..., and t+m-1. The input port searches the tokens for the HOL packets. The input port searches the token for time t first. If no token can be found, the input port searches the token for time t+1; and so on. An example is given in Figure 3-8 to illustrate the operation.

For easy discussion, assume the switch size is 4 x 4. At time t-1, four streams of tokens pass to the input ports. Since the HOL packets at input port 0 and input port 1 are both destined to output port 1, the HOL packet at input port 0 reserves token 1 for time t and the HOL packet at input port 1 reserves token 1 for time t+1. At time t, the HOL packets that have been assigned transmission time t are sent out. In the mean time, four streams of tokens pass to the input ports. Notice that the token streams for time t expire and the token stream for time t+4 is joined. Since the HOL packet at input port 1 has been assigned a transmission time, input port 1 reserves a token for the packet behind the HOL packet. This is referred as queue bypass scheme. With future scheduling and queue bypass scheme, the switch throughput is largely increased. (If m is equal to N, the throughout can be at least the same as in Reference 3-5.) Note that the tokens can be reused multiple times until the tokens expire. The same procedure repeats at time t+1.

The new scheme combines the advantages of centralized reservation scheme and future scheduling, i.e., easy implementation and high throughput. Also, unlike the scheme proposed in Reference 3-5, this new scheme will not transmit packets out-of-sequence.



a) Time t-1



b) Time t

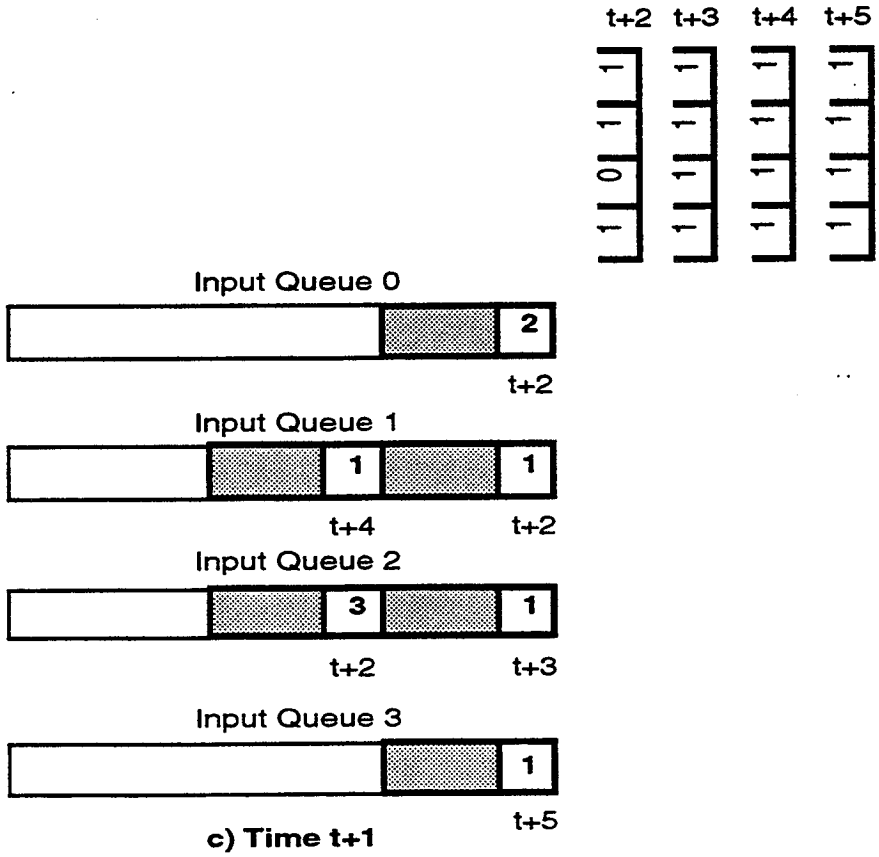


Figure 3-8: An Illustration of the New Output Reservation Scheme

3.1.2.9 The Proposed Output Port Reservation Scheme

Centralized scheme without pipeline operation must complete sending the token streams (multiple times considering priority and a large checking depth) to all N input port in one slot time. The reservation speed gets higher when size N gets larger. Centralized scheme with pipeline operation sends token streams in a parallel, pipeline fashion to different input ports. The reservation speed is independent of the switch size. However, the pipeline mode introduces more hardware complexity. Since the on-board switch size is not large, the advantage of using the centralized scheme with pipeline operation is not significant. The first proposed scheme is to use the basic centralized ring reservation scheme. Transfer of a multicast packet at the input port has the call splitting capability. Using the centralized ring reservation scheme, a larger checking depth is required to achieve a high throughput.

The second proposed scheme is to use the centralized ring reservation scheme with future scheduling. The number of token streams, which can be sent in parallel, is limited

by hardware complexity. This scheme can increase the switch throughput without increasing the switch speed; consequently, no output queueing is required.

Distributed scheme can largely increase the switch throughput without increasing the switch speed. In addition, the distributed scheme may be made more reliable than a centralized scheme. However, due to the limited budget, the performance of the distributed scheme can not be analyzed at this point of time. Based on the above discussion, two output reservation schemes are proposed:

- Centralized input ring reservation scheme with a large checking depth and without future scheduling
- Centralized input ring reservation scheme with a large checking depth and with future scheduling

The final selection will be determined at the "High Level Design" task.

3.2 Satellite Virtual Packets

Satellite virtual packet (SVP) concept has been proposed for unified routing, control and management within the satellite B-ISDN [2-1]. Unicast SVPs are created by prepending a header to one point-to-point ATM cell or a group of point-to-point ATM cells destined to the same downlink beam (or the same receiving earth station). For point-to-multipoint ATM cells, a cell or a group of cells destined to the same set of downlink beams are grouped into a multicast SVP. Formatting cells into SVPs at the earth station can avoid on-board VPI/VCI processing and HEC processing, simplify the space segment complexity without introducing much hardware at the earth station, and bit interleaving of cell headers can be naturally achieved [2-1]. Although, originally, SVPs are proposed to accommodate only the ATM cells, the SVP payload can be extended to support other other high-speed/wideband traffic such synchronous digital hierarchy (SDH) and synchronous optical network (SONET). (SVP can also support low-rate traffic such as frame relay and consultative committee for space data systems [CCSDS]). SVPs are served as a multi-media container within the satellite network. Since cells are already in packet format, to place the cells in the SVP payload, no segmentation is required. For SDH and SONET, a segmentation protocol is necessary to segment the signal into blocks. After segmentation, a sequence number (SN) and a virtual channel number (VCN) are required in the SVP header. The SN is for reassembly purpose and the VCN is to identify the connection. Although SN and VCN are not necessary for the SVPs with cells as payload, to have a unified header structure, VCN and SN are suggested to be placed in every SVP header. Note that the payload of one SVP does not support mixed protocols. For example, the SVP payload either consists of ATM cells or SDH circuit slots, but not both.

Three important topics are discussed in this subsection. The first one deals with whether VPI/VCI processing for ATM traffic is necessary for the FPS. The second addresses the SDH packetization procedure at the earth station. The third addresses the alternatives of SVP formats and the required modifications for the FPS to accommodate the SVPs.

3.2.1 VPI/VCI Processing for ATM Traffic

For ATM cells, 24 bits VPI and VCI at the user network interface (UNI) and 28 bits at the network node interface (NNI) are available for routing information. VPI can have either a local or a global significance within the satellite network based whether the VPI is unique in the satellite network. If VPIs are unique within the satellite network, VPI has a global significance. If VPIs can be reused at different nodes (terminals and on-board switch), VPIs only have a local significance. The advantages and disadvantages of VPI with a local significance and a global significance are discussed below.

a. Topology flexibility: VPI with a local significance provides more flexibility in adapting the network topology for network expansion (such as node addition and link addition) and node failure. In contrast, VPI with a global significance has less flexibility. For example, once a node fails, it may be hard to reuse the VPI space reserved for the node.

b. Node processing and delay: Since VPIs with a global significance have a unique VPI for each NNI, the transit node only performs routing while VPI retranslation is not necessary. The result is that node processing cost is small and the call setup delay is minimal.

c. Addressing space: The addressing space for VPIs with a global significance is shared by the NNIs within the satellite network. This may not be feasible for a large network since the number of NNIs is large.

From the above discussion, the VPI with a local significance is more advantageous than the VPI with a global significance in terms of flexibility and the addressing space. However, the VPI with a local significance requires the VPI to be translated at every VP terminator (such as a switch). Since the space segment contains an FPS, VPI retranslation on-board will result in a larger delay, higher processing cost, and more memory requirements. By using SVPs, VPI with a local significance can be adopted and no VPI retranslation on-board is necessary. The reason is that cells becomes the payload of SVPs, and SVPs can be routed through the FPS using the routing tag and the connection can be identified using the VCN.

In summary, the VPI with a local significance is used within the satellite network. The VPI needs to be retranslated at the earth stations. However, no VPI translation is required at the on-board FPS.

3.2.2 SDH Packetization

SDH and SONET will be used to support B-ISDN traffic in Europe and Northern America, respectively. SDH (or SONET) supports both ATM connections and circuit switched connections. The SDH signal is converted into SVP format at the earth station.

The SDH signal can be divided into SDH information payload and overheads. If the SDH payload contains cells, then the location of the first byte of the virtual container (VC-4) path overhead is indicated by the AU pointer. The VC-4 consists of a container (C-4) and the path overhead. The cells can be extracted from the C-4 container by processing the H4 offset within the path overhead. After his, the cells will be grouped into SVPs.

If the SDH payload contains circuits, then the circuit slots are directly put into the SVP payload and designate one field in the SVP header (payload type field) to identify that this packet consists of circuit slots. These SVPs can be treated as circuit data, which exhibit periodic and deterministic natures. Integration operation of circuit and packet switched traffic using a FSP is addressed in Section 3.4.

The SDH contains standard overhead bits for operation, maintenance, communication, and performance monitoring functions. These overheads consist of section overhead (SDH), path overhead (POH), and AU-4 pointer. These overheads are placed directly into the SVP payload. The destination information for these circuit slots is contained from a separate signaling channel (such as SS7). This information can be used to format the routing tag at the earth station.

3.2.3 The SVP Format

The size of the SVP is the first issue which needs to be considered for the SVP format. Since the size of the SVP header will be fixed, the more the cells (or bits) are put into the SVP, the higher the transmission efficiency will be. A larger SVP also increases the packet interarrival time, which lightens the speed requirement for the FPS. However, a larger packet will have a longer packetization delay at the earth station and the payload efficiency is low if not enough number of cells can be filled into the packet. A larger packet size will result in larger buffer requirement, longer end-to-end delay, and worse delay jitter. A larger packet will also increase the number of bits of the SVP payload in error. No optimal size can be found for single-size SVPs. Since the buffer requirement is the main concern for the space segment and the delay determines the quality of circuit emulation service, it is envisioned that one SVP should only contains several cells.

Based on the performance analysis of SVP (supporting only ATM cells) transmission through the earth station, to fully utilize the SVP concept without affecting the delay quality, the uplink and downlink has to operate at a very high utilization (80%) and the single-size SVP should be kept small (≤ 4 cells). The 80% link utilization defines the lower bound of the FPS throughput.

We may wonder if a SVP cannot be filled up with cells, it simply means that there is not enough traffic in the system. Therefore, padding of idle cells in a SVP is not considered to be bandwidth inefficient. This statement is true only if there is one type of service and one beam. The satellite bandwidth is shared by different terminals in different beams and by different types of services. Assume the loading distribution for different downlink beams is not uniform for the terminals in the same uplink beam. Then the incoming cells destined to the heavily loaded downlink beam have high utilization and those destined to the lightly loaded downlink beam have low utilization. If idle cells are constantly inserted into the SVPs destined to the lightly loaded downlink beam, some bandwidth is wasted. Consequently, the amount of bandwidth, which can be utilized by the SVPs destined to the heavily loaded downlink beam, is reduced. Also bandwidth saving on one type of service can be used by another type of service. For example, the uplink beam supports packet switched traffic and circuit switched traffic. Then bandwidth saving on packet switched traffic can be used for circuit switched traffic. Therefore, increasing the SVP transmission efficiency and SVP payload efficiency is vital to the bandwidth limited satellite environment. If there is indeed not enough traffic in the system, idle SVPs will be sent from the terminals to maintain the frame synchronization.

Since no optimal size can be found for SVPs, to efficiently support ATM cells, SDH, and other traffic within the satellite network, SVPs may have to use multiple sizes. The SVPs with multiple sizes concept is explained below. Assume that one single ATM cell size is chosen as the basic SVP information payload size. Then the satellite network may support single-cell SVPs, two-cell SVPs, four-cell SVPs, and so on. The overall processing requirement to support SVPs with a limited set of sizes compared with that to support the variable-size packets is reduced since the SVP size can be determined using a size indication field (not the packet length field). The payload efficiency is increased for SVPs with multiple sizes since the SVP has the flexibility of using different payload sizes to

accommodate traffic with different intensities. The transmission efficiency is high if a large payload size is used and is low is a small payload size is used. Overall, the transmission efficiency for the SVP with multiple sizes is about the same as that for single-size SVPs. In summary, employing SVPs with multiple sizes combines the advantages of the fixed-size packet and the variable-size packet.

A timer is associated with each SVP. If the timer expires, the SVP will be sent out with the cells (or bits) currently in the SVP. If the current number of cells is not equal to one of the SVP sizes, there are two options. These two options are explained using an example. Assume a SVP has four sizes: one cell, two cells, four cells, and eight cells. Let the current number of cells in the SVP is 3. Option 1 is to send out the SVP as a 2-cell SVP. The remaining cell inserts into the next SVP. Option 2 is to send out the SVP as a 4-cell SVP. Since there are only three cells in the SVP, the SVP is padded with an idle cell.

There are two options for the SVP header format depending on grouping of cells (or bits) into a SVP is based on the downlink beam or the receiving earth station.

We first discuss the SVP header format when grouping of cells (or bits) into a SVP is based on the downlink beam. This grouping method is more applicable for a very small aperture terminal type (VSAT-type) satellite network with a few spot beams and a large number of terminals in each beam. Broadcast connections can be achieved easily for the receiving terminals within the same downlink beam since no packet duplication is necessary. It is proposed that the header consists of the following fields: the switch routing tag, size indication field, quality of service (QOS) field, control field, payload type (PT) field, virtual channel number (VCN), sequence number (SN), and forward error control (FEC) field (see Figure 3-9).

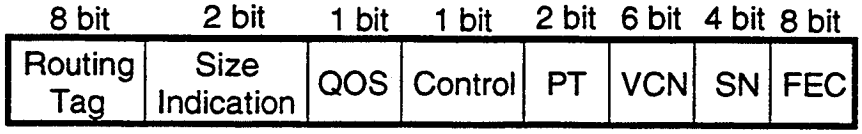


Figure 3-9: Tentative SVP Header Option 1

The routing tag is to identify the downlink beam and is also used for routing through the on-board switch. The size and format of the routing tag depends on the fast packet switching architectures and connectivity. For phase 2 implementation, only multicast FPSs are considered. One possible routing tag format uses a series of 1's and 0's. The position of all the 1's means all the destined output ports. The size of the routing tag is the same as the switch size. The routing tag is proposed to be prepended at the earth station. An inherent advantage of this approach is that fault-reconfiguration of the FPS can be easily achieved by simply changing the routing tag of the packet at the earth station. As a result, the packet can be routed through a fault-free path within the switch. The size indication field is to identify the size of the SVP (for example, single-cell SVP, 2-cell SVP, and so on). The QOS field is for QOS control. All the cells within the SVP should have the same QOS. For example, all the cells within the SVP will have the same cell loss priority (CLP). Based

on the QOS field, the FPS can perform priority control to guarantee the QOS for certain services, and also drop the low-priority cells if congestion occurs. The control field is to identify the SVP is an information packet or a control packet. If the SVP is a control packet, it will be routed to the OBC. Also the OBC will generate control SVPs (containing congestion status), and these packets will be broadcasted to all the earth stations. The payload type is used to identify whether the payload consists of cell, circuit slots, or other types of traffic. The SN is for reassembly purpose and the VCN is to identify the connection. The FEC field is used to correct and detect errors in the SVP header. The FEC can also be used for synchronizing the SVPs as the cell delineation algorithm performed in the ATM cell synchronization procedure. SVP synchronization procedure using the FEC field will be presented latter. Due to the complexity of FEC coder/decoder, the FEC may leave as an option for Phase 2 development.

The preliminary sizes chosen for the fields are the routing tag 1 byte, the size indication field 2 bits, QOS field 1 bit, control field 1 bit, payload type field 2 bits, VCN field 6 bits, SN field 4 bits, and FEC field 1 bytes. The size of the SVP header has 4 bytes.

Note since cell (or bit) grouping is performed based on the destined downlink beam, the source station and destination station addresses are not necessary (when VCN has a global significance). To extract the cells, the circuit slots, or other types of traffic in the same SVP destined to the earth station, the following procedure must follow. First, the earth stations within the same downlink beam use the payload type field to identify whether the payload consists of cells, circuit slots, or other types of traffic. Then the earth stations use the VCN to identify the connections for the type of traffic. The VCN has a global significance within the satellite network. The sequence number is used to reassemble the original packet format. The VCN for each type of service is set up at the signaling phase.

There are two alternatives for a single-size SVP containing only one cell. The first alternative is to have a header in front of every single cell. The SVP payload consists of 53 bytes. The SVP format is shown in Figure 3-10A. Clearly, the size indication field is not required in the SVP header. This approach has the simplest implementation but it also has the largest overhead. The second is to map the five-byte cell header into the SVP header at the sending station. The SVP format is shown in Figure 3-10B. The SVP payload consists of 48 bytes. The on-board switch routes the SVP based on the routing tag in the SVP header. When the receiving station receives the SVP, the SVP header is mapped back to the cell header. Although mapping functions are required at the earth station, this approach has the smallest overhead (for ATM traffic).

Although the single-size SVPs containing one cell has the highest transmission efficiency for ATM traffic, it is not efficient for SDH traffic. Depending on the traffic pattern, multiple sizes of SVPs may be required. Before discussing the format of multiple sizes of SVPs, the "basic SVP size" has to be determined. The basic SVP information payload is chosen to be 53 bytes to conform with the ATM cell size. The "basic SVP" consists of the basic information payload and the header, and its size is 57 bytes. There are two alternatives for a large SVP. The first alternative is that a larger SVP size is multiple of the basic SVP size. The second alternative is that the information payload size of a larger SVP is multiple of the information payload size of the basic SVP. These two alternatives are shown in Figure 3-11.

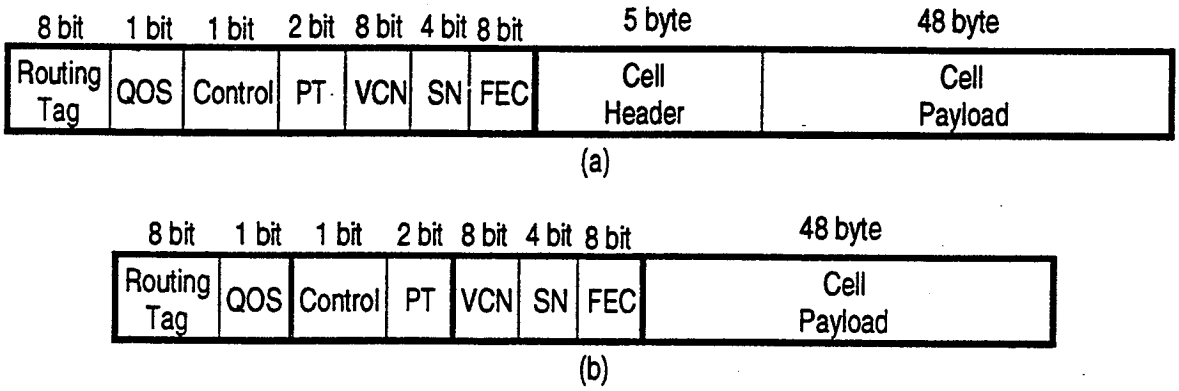
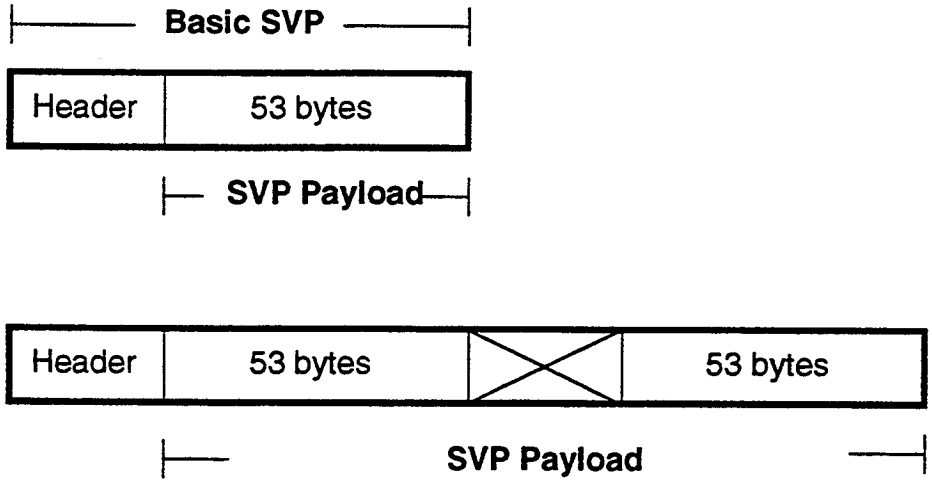
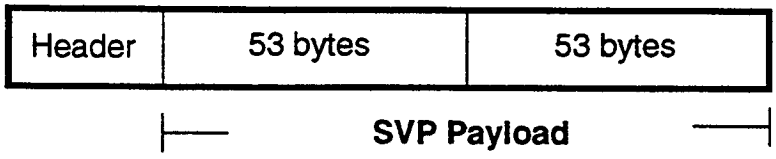


Figure 3-10: Two Alternatives for Single-Size SVPs Containing One Cell



Alternative 1 for a 2-Cell SVP



Alternative 2 for a 2-Cell SVP

Figure 3-11: Two Alternatives for a 2-Cell SVP

Since a large SVP size in the first alternative is exactly multiple of the basic SVP size, the uplink and downlink transmission format can use the TDM slotted-mode, where a single slot corresponds to the basic SVP size. In the second alternative, the uplink and downlink are unchannelized. The second alternative is more advantageous only if the on-board processor has the capability of segmenting a large SVP into several basic SVPs; in this case, the on-board processor can operate at single-slotted mode, where one slot corresponds to one basic SVP.

These two alternatives of supporting multiple SVP sizes are tabulated in Table 3-1A and Table 3-1B.

Table 3-1A: SVP Sizes Alternative 1 for SVP Header Option 1 (Scenario A)

	SVP size	information payload
1-cell SVP	57 bytes	53 bytes
2-cell SVP	114 bytes	110 bytes
4-cell SVP	228 bytes	224 bytes
8-cell SVP	456 bytes	452 bytes

Table 3-1B: SVP Sizes Alternative 2 for SVP Header Option 1 (Scenario B)

	SVP size	information payload
1-cell SVP	57 bytes	53 bytes
2-cell SVP	110 bytes	106 bytes
4-cell SVP	216 bytes	212 bytes
8-cell SVP	428 bytes	424 bytes

The SVP size alternative 1 with SVP header option 1 is referred to scenario A. The SVP size alternative 2 with SVP header option 1 is referred to scenario B. For scenario A, the simplest on-board operation is to segment the SVP into multiple basic SVPs and the routing tag of the SVP is prepended in front of each basic SVP. The on-board switch is still operated in single-slotted mode, where one slot corresponds to one basic SVP plus the routing tag. The output port reassembles the basic SVPs back to the original SVP. All the routing tags are removed before reassembly. (Note for the crossbar switch, the routing tag

is not required to be sent with the packet during transmission, since the crossbar switch is centralized control not self-routing. In this case, the output port simply reassembles the packets back to the original SVP. No removal of routing tags is necessary.) For scenario B, the simplest on-board operation is to segment the SVP payload into multiple basic SVP payloads and the header of the SVP is prepended in front of each basic SVP payload. All the basic SVP headers are removed before reassembly except the header of the first basic SVP. After the SVP has been formed, the output port sends the SVP to the output transmission link.

The disadvantage of the above operations is that the transmission of a large SVP (containing several basic SVPs) through the switch can not be guaranteed to be continuous. The reason is that output contention result among different input ports is random unless some special mechanism or priority control is applied. The proposed output contention control scheme for multiple sizes of SVPs is presented in Section 3.2.4.

If grouping of cells (or bits) into a SVP is based on the receiving earth station, the additional information required for the SVP header is the receiving earth station ID. The grouping method is more applicable to a satellite network with a few larger earth stations. The receiving earth station ID is to identify the receiving earth station within the same downlink beam. It is proposed that the SVP header consists of the following fields (see Figure 3-12): the switch routing tag, size indication field, receiving earth station address, quality of service (QOS) field, control field, payload type (PT) field, virtual channel number (VCN), sequence number (SN), and forward error control (FEC) field.

8 bit	8 bit	2 bit	1 bit	1 bit	2 bit	6 bit	4 bit	8 bit
Routing Tag	Receiving ES ID	Size Indication	QOS	Control	PT	VCN	SN	FEC

Figure 3-12: Tentative SVP Header Option 2

The preliminary sizes chosen for the fields are: the routing tag 1 byte, the receiving earth station address 1 byte, the size indication field 2 bits, QOS field 1 bit, control field 1 bit, payload type field 2 bits, VCN field 6 bits, SN field 4 bits, and FEC field 2 bytes. The size of the SVP header has 6 bytes. Therefore, the basic SVP size consists of 59 bytes.

As discussed before, there are two alternatives for the size of a larger SVP. The sizes supported by the two alternatives are listed In Tables 3-2A and 3-2B.

The SVP size alternative 1 with SVP header option 2 is referred to scenario C. The SVP size alternative 2 with SVP header option 2 is referred to scenario D.

Table 3-2A: SVP Size Alternative 1 for SVP Header Option 2 (Scenario C)

	SVP size	information payload
1-cell SVP	59 bytes	53 bytes
2-cell SVP	118 bytes	112 bytes
4-cell SVP	236 bytes	230 bytes
8-cell SVP	472 bytes	466 bytes

Table 3-2B: SVP Size Alternative 2 for SVP Header Option 2 (Scenario D)

	SVP size	information payload
1-cell SVP	59 bytes	53 bytes
2-cell SVP	112 bytes	106 bytes
4-cell SVP	218 bytes	212 bytes
8-cell SVP	430 bytes	424 bytes

3.2.3 SVP Acquisition and Synchronization

The SVP acquisition and synchronization can be achieved using three schemes. The first scheme follows the synchronization method used in the frame relay. The second scheme follows synchronization method used in the TDM frame synchronization and a frame format is required. The third scheme follows the techniques used in the ATM cell header error control synchronization (ATM cell self-delineation) and no external frame format is required.

Scenarios B and D should use scheme 1 or scheme 2. In this scheme, two SVPs are separated by closing and opening flags, similar to the high-level data link control (HDLC) flags. The disadvantage of this approach is that no duplication of the flag pattern is allowed in the SVP. Therefore, bit stuffing/destuffing adds more complexity to the FPS. The second scheme is to use a fixed size frame with a frame marker. The synchronization scheme is very similar to ATM cell self-delineation procedure except the frame synchronization uses a prestored unique word to search for the frame marker and ATM uses the syndrome of the decoder to search for the cell header. In this procedure, the byte boundary of the frame has to be established. Bit-by-bit searching is performed until the first byte of the unique word is found. When the first byte is found, the next multiple bytes are used to match with the unique word. If they do not match, the procedure enters bit-by-bit

searching. If they match, the next n consecutive frame markers are compared with the unique word. If they all match, frame synchronization is achieved. The frames are always in synchronization unless m consecutive frame markers do not match with the unique word. If this occurs, the procedure enters bit-by-bit searching. If one of the n frame markers does not match, the procedure enters bit-by-bit searching. Since there are multiple sizes of SVPs, some inefficiency may arise if the SVPs can not fit perfectly into a frame.

Scenarios A and C should use scheme 2 or scheme 3. As previously discused, since there are multiple sizes of SVPs, some inefficiency may arise if the SVPs can not fit perfectly into a frame. Scheme 3 does not have the disadvantage of scheme 2. Scheme 3 is discussed below. Since the SVP size is not fixed, the synchronization procedure used for ATM cell header has to be modified such that the size indication information is included in the synchronization process. The SVP delineation method is to use the correlation between the 1-byte FEC and the 3-byte header (in Scenario A). There are three states in the SVP delineation state diagram: HUNT, PRESYNC, and SYNC (see Figure 3-13). The HUNT state is used to search for the SVP header. The PRESYNC state is to verify that the header found by the HUNT state is correct. The SYNC state is to maintain synchronization of the SVP stream.

The implementation of the HUNT state for the multiple-size SVPs follows the same design principle as discussed in [3-7]. In this state, the receiver searches for the SVP header by using a correlation between the 3-byte header and the 1-byte FEC field. A bit-by-bit search is used to identify the SVP boundary. When the first SVP boundary is found, the location of the SVP is recorded and the second SVP boundary is searched. When the second SVP boundary is found, the location of the second SVP is also recorded. The interval between the first SVP and the second SVP is calculated. If the interval is multiple of the size of the basic SVP, the initial acquisition is achieved and the state enters the PRESYNC state.

When the receiver is in the PRESYNC state, the receiver will verify the SVP header using the correlation between the SVP header and the FEC field. At the same time, the receiver will extract the SVP size indication field to determine the size of the SVP. By doing this, the receiver can identify the start of the next SVP, i.e., identify the boundary of SVPs. If there is one incorrect SVP header within the next n SVPs, the state will return to the HUNT state, If the next n SVP headers contain no errors, the state enters the SYNC state. In this state, the decoder performs error correction. The state will stay in SYNC until m incorrect SVP headers are found. If m incorrect SVP headers are found, the state will return to the HUNT state.

The selection of the synchronization scheme is determined at the "High-Level Design" task.

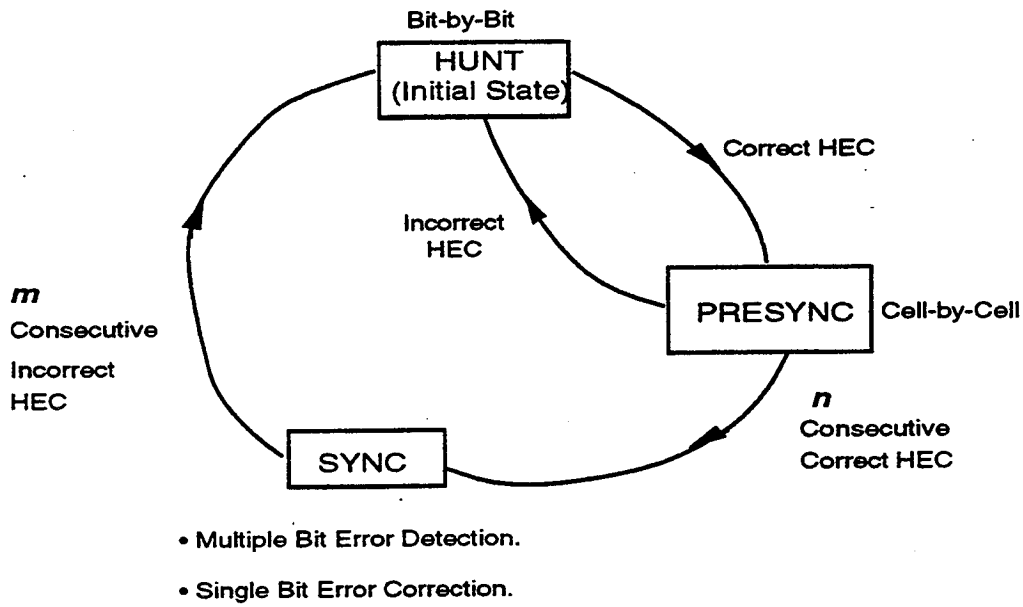


Figure 3-13: ATM Cell Header Error Control Synchronization

3.2.4 Switch Operation for Multiple Sizes of SVPs

From the discussion in Section 2, the multicast crossbar is chosen to be the switching architecture for Phase 2 development. The necessary modification of the switch operation to accommodate the SVPs with multiple sizes is in the output contention resolution. From the discussion in Section 3.1, the output port reservation scheme is most preferable for the multicast crossbar. The following discusses the output port reservation scheme for a crossbar switch to accommodate the SVPs with multiple sizes.

3.2.4.1 Point-to-Point Output Port Reservation

In this switch configuration, the operation of the switch has to be able to operate in single-slot mode and multiple-slot mode concurrently. To facilitate the discussion, assume the SVPs consist of single-cell SVPs (or basic SVPs) and two-cell SVPs only.

The operation for scenarios A and C is discussed first. The switch is operated on slotted-mode, where the slot size is the same as the size of a basic SVP . If the incoming packet is a 2-cell SVP, then the operation of the switch has to be modified to accommodate that the packet size is larger than the slot size. For scenarios A and C, the input port has to be able to reserve two contiguous slots in advance in order to successfully transmit the 2-cell SVP. This modification can be achieved easily with the centralized ring reservation scheme. There are three different implementations. The first implementation is that the token generator generates two streams of tokens: one stream is for the next slot and one stream is for the following slot. Each token represents one output port. If the incoming packet is a basic SVP, the operation of the on-board switch is normal. The basic SVP only

processes the token at the first token stream, i.e. the next-slot token stream. To avoid out-of-sequence transmission, the 2-cell SVP has to reserve two tokens in two different streams for the same output port simultaneously to successfully transmit the SVP. During transmission, the input port prepends the routing tag to each basic SVP within the same SVP. (Note for the crossbar switch, no routing tag is required for the SVP during transmission.) Due to the routing tag overhead, the switch has to operate at a higher speed than the link speed.

The second implementation is to use only one token stream. However, the input port has the capability of keeping the token for multiple slots. If the input port needs to transmit a 2-cell SVP, it has to seize the token for two time slots once the first token for the 2-cell SVP is reserved. The output reservation is executed only once for the first basic SVP of the 2-cell SVP. When the input finishes transmitting both of the basic SVPs in the SVP, the input port releases the token.

The third implementation is to allow both of the basic SVPs in the 2-cell SVP participate output reservation. However, priority control is applied such that both of the basic SVPs in the 2-cell SVP can be transmitted contiguously. The priority is given to the second basic SVP of the 2-cell SVP, when there is output contention between a single-cell SVP and the second basic SVP of the 2-cell SVP. However, the first basic SVP of the 2-cell SVP should have the same priority as a single-cell SVP. Note it is not possible to have contention between two second basic SVPs belonging to two different 2-cell SVPs. The design considerations for priority control are discussed in Section 3.3.

When the first basic SVP arrives to the output port, the output port examines the size indication field and allocates the memory space. When the successive basic SVPs coming to the output port, the basic SVP payload is placed into the proper location for reassembly. When all the basic SVPs within the SVP have been received, the SVP can be transmitted to the output link.

The operation for scenarios B and D is to segment a 2-cell SVP into two single-cell SVPs on-board. After segmentation, the header of the 2-cell SVP is copied to the second basic SVP. Due to the header overhead, the switch has to operate at a higher speed than the link speed. The output port examines the size indication field and performs assembly if necessary. In this situation, it is possible that two basic SVPs belonging to the same SVP do not come into the output port contiguously. This discontinuity complicates the design of the output buffer. The discontinuity can be avoided if priority control is executed. Priority should be given to the second basic SVP of the 2-cell SVP when the second basic SVP is contended with other types of SVPs.

Since the on-board switch is operated in single slotted-mode, the on-board processor operation for scenarios B and D will not be discussed any further.

3.2.4.2 Point-to-Multipoint Output Port Reservation

Since the switching fabric is nonblocking, the output port reservation scheme used for point-to-point connections can also be used for point-to-multipoint connections with a slight modification. For point-to-multipoint connections, each input port can reserve more than one and up to N output ports at a time.

The operation of the multicast switch for SVPs with different sizes is complicated by the multicast connection. It is very unlikely that all the multicast basic SVPs in the same SVP can reserve all the output ports in advance; hence, the output reservation process proposed for the point-to-point case has to be modified. As shown in Table 3-3, there are four possible operations for an input port to handle a multicast packet.

Table 3-3: Four Possible Operations for Input Port to Handle a Multicast Packet

	operation 1	operation 2	operation 3	operation 4
call splitting	yes	yes	no	no
continuity	yes	no	yes	no

If call splitting is allowed for the multicast packet at an input port, it means a multicast packet can finish its transmission in multiple slots. If continuity is a requirement, it means the basic SVPs within the same SVP have to transmit to the output ports contiguously. These different operations are explained below.

Assume a multicast 2-cell SVP is destined to output ports 0, 1, 4 and 6. For operation 1, a multicast basic SVP can finish its transmission in multiple slots. However, the first basic SVP and the second basic SVP have to be transmitted to the same set of output ports contiguously. Assume that the first basic SVP successfully reserves output ports 0 and 6 at slot S, then the second basic SVP should also reserve output ports 0 and 6 at slot S+1. If this condition is achieved, the SVP can be transferred to outputs 0 and 6. The transmission of the basic SVP to destinations 1 and 4 can be accomplished latter.

For operation 2, a multicast basic SVP can finish its transmission in multiple slots, and the first basic SVP and the second basic SVP do not have to be transmitted to the same set of output ports contiguously. Assume that the first basic SVP successfully reserves output ports 0 and 6 at the slot S, then the second basic SVP, to avoid out-of-sequence, can at most reserve output ports 0 and/or 6 (but not 1 and 4) at any slot T, where T > S. The second basic SVP can be transmitted to output ports 1 and 4 only after the first basic SVP has finished its transmission to the output ports 1 and 4.

For operation 3, a multicast basic SVP must finish its transmission in one slot, and the first basic SVP and the second basic SVP have to be transmitted to the same set of output ports contiguously. In this case, the basic SVP has to reserve output ports 0,1,4, and 6 at the same slot and the second SVP should also reserve output ports 0,1,4, and 6 at the next slot. If this condition cannot be achieved, the SVP cannot be transmitted.

For operation 4, a multicast basic SVP must finish its transmission in one slots, and the first basic SVP and the second basic SVP do not have to be transmitted to the same set of output ports contiguously. In this case, the first basic SVP must reserve output ports 0,1,4, and 6 at the same slot (S) and the second basic should also reserve output ports 0,1,4,

and 6 at same slot (T), where T > S. However, the first basic SVP transmission time and the second basic SVP transmission time does not have to be contiguous.

3.2.5 The Proposed SVP Formats

The header of SVPs contains the routing tag and other satellite network internal fields such as payload type and QOS. The routing tag is used to route through the on-board switch. The routing tag is inserted in the SVP header at the earth station. For ATM application, The VPI has a local significance in the satellite network. The VPI needs to be retranslated at the earth station. However, no VPI retranslation is required at the on-boar switch. Grouping of cells (or other types of traffic) should be based on the downlink beam if there are a large number of terminals in the network. Grouping of cells (or other type of traffic) should be based on the receiving earth station if there are a few, large earth stations in the network. If single-size SVP is chosen for Phase 2 development, the SVP size should be less than or equal to that of 4 cells. The formats of the two alternatives for single-size SVPs are shown in Figure 3-14. If the traffic foreseen is very diverse, then multiple-size SVPs should be considered. There are four different sizes: single-cell SVP, 2-cell SVP, 4-cell SVP and 8-cell SVP. The multiple size SVP header is shown in Figure 3-9 and the SVP format is shown in Figure 3-11 (Alternative 1). For multicast multiple-size SVPs, the switch operation should allow call splitting and enforce continuous transmission for the SVP packet through the switch. Final selection of SVP format is determined at the "High Level Design" task.

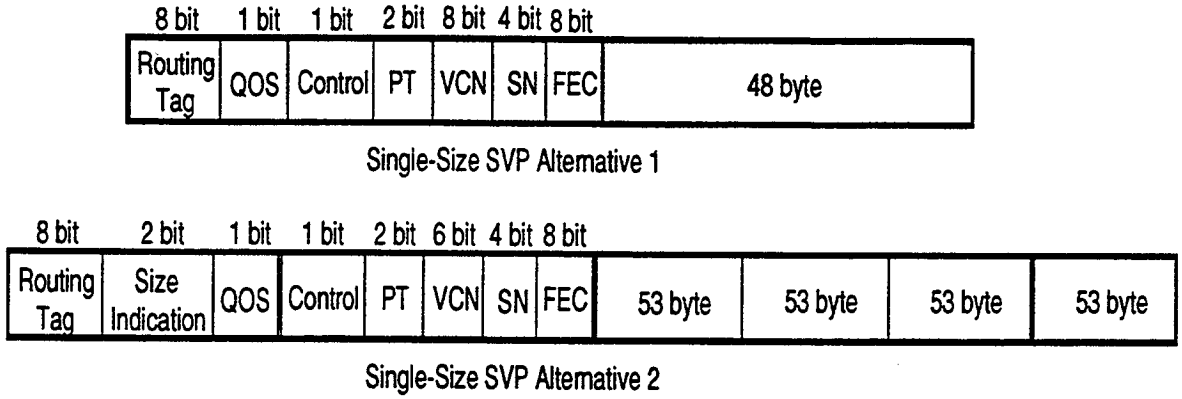


Figure 3-14: The Proposed Single-Size SVP Formats

3.3 Priority Control

This section investigates how to effectively implement priority control for the multicast crossbar switch. There are two forms of priority control: priority control for scheduling the packet transfer at a switch and priority control for congestion [3-8]. Priority control for scheduling the packet transfer in a switch is discussed using the output port reservation scheme presented in Section 3.1. In congestion control, low-priority packets are dropped before high-priority packets in case of congestion to minimize the influence and to maintain the QOS of higher priority connections. Priority for congestion control will be discussed in the task " Critical Element Design and Simulation".

Researchers use two approaches to tackle the priority control (for ATM): switch throughput and buffer management. By properly dividing the switch throughput among packets with different priorities, the QOS of high priority packets can be guaranteed. By careful design of the buffer, low priority packets will be dropped before the high priority packets. Subsection 2 presents the priority control schemes (proposed in the past) used in a multicast crossbar switch and the implementation issues. Subsection 3 presents the buffer management scheme at each input port and a new output port reservation scheme. The new output port reservation scheme has the capability of adjusting the QOS of each connection based on its priority. The salient feature of the new output port reservation algorithm is that when the QOS of the high-priority packets has been set to the desired value, the QOS of the low-priority packets can be adjusted without affecting the QOS of the high-priority packets. The recommended priority control scheme for Phase 2 development is presented in Subsection 4.

3.3.1 Different Priorities

To provide different levels of quality of service (QOS) for different classes of services, priority control on the fast packet switch is necessary. In general, there are two parameters associated with the QOS of a packet: the time priority and the loss priority. The time priority is used to distinguish real time traffic (such as circuit switched data) and non-real time traffic (such as packet switched data). With time priority, the packet transfer delay (PTD) and packet delay jitter (PDJ) of high-priority packets are reduced at the expense of low-priority packets. The loss priority is used to distinguish loss sensitive data and loss insensitive data (such as datagram). With loss priority, the packet loss ratio (PLR) of the high-priority packets is reduced at the expense of low-priority packets. For ATM cells, priority control is performed using the cell loss priority (CLP) bit in the ATM cell header. The CLP bit can provide priority control for two classes of services. If more than two classes of services are supported, then the reserved (RES) bit in the ATM cell header can also be used for priority control. Satellite virtual packets (SVPs) are created by prepending a header to a group of cells (or other types of data) destined to the same downlink beam for unified routing, control, and management. The formats of SVPs have been presented in Section 3.2. The QOS field in the SVP header can be used to segregate the traffic into different classes, and different levels of control can be applied to different classes.

3.3.2 Priority Control using the Centralized Ring Reservation Scheme

Priority is used to meet different packet loss ratio requirements and switch delay requirements for different services. A high-priority packet is guaranteed to win the output contention resolution when it is contended with other low-priority packets. The centralized ring reservation scheme has been chosen to be one of the output contention resolution schemes in Section 3.1. Only one type of priority is considered. It is based on one of the QOS requirements (such as packet loss ratio).

Priority control for a multicast switch with nonblocking switching fabric, input buffering and centralized ring reservation scheme to resolve output contention was discussed in Reference 3-9. Two approaches were proposed to implement the priority control.

The first approach, referred as "overwrite" scheme, is to modify the token format. The priority level is included in the tokens. A typical format of tokens is shown in Figure 3-15. There are N tokens for N output ports and there are N priority subfields for N tokens. Whenever an input port reserves the output port, the priority of the packet waiting in the queue is inserted in subfield P_i, where i is the position of token i. Following the centralized ring reservation algorithm, each input port tries to reserve the output port by examining the output port availability. If the output port has been reserved, then the input port also checks the priority level associated with this token. If the priority level is equal to or higher than its own priority level, then no action. If the priority level is lower than its own priority level, the input port uses its own priority level to "overwrite" the priority subfield. If this occurs, the input port whose priority subfield has been overwritten needs to be notified. The authors in Reference 3-9 did not describe how to implement the notification. In fact, the notification scheme is straightforward. After all the input ports have finished reservation, the token stream is sent back to the input ports one more time for confirmation. Every input port checks the priority subfield associated with the token to see if the priority is still the same as its own priority. If they are the same, confirmation is achieved and the packet can be transmitted at the beginning of the next slot. If they are different, it means some higher priority packet at another input port has overwritten the token for the particular output port. The result is that the low-priority packet has to retry the reservation request at the next slot. In summary the first scheme loops the tokens through the input ports twice. Loop 1 is for the input ports to reserve the output ports and loop 2 is used for the input ports to confirm that the reservation of the output ports is successful.

The second scheme, referred as "loop" scheme, is to send the tokens to the input ports k times , i.e., to loop through the input ports k times, where k is the number of priority levels. Starting from priority k (the highest priority), only the packets with priority k can reserve the output ports at this run. The next run is for priority k-1 packets and so on. With this particular scheme, the priority subfield in the token stream is not necessary since the priority control is performed on different priority levels of packets at different runs. Consequently, modification of the token format is not needed, i.e., the token format only consists of the output port bit map. This scheme guarantees that a high priority packet wins over a low priority packet during output port reservation. If two packets have the same

priority, the resolution is based on the sequence of token passing. Evidently to achiever fairness, the start of the token stream must be alternated.

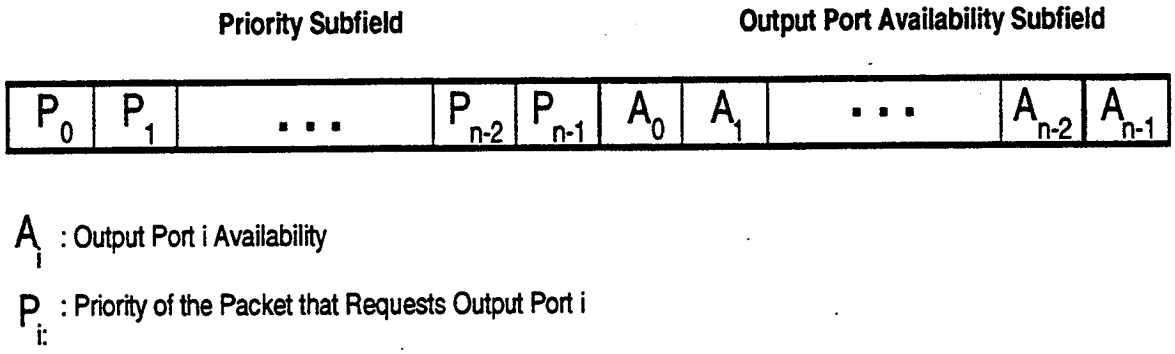


Figure 3-15: Token Format With Priority Subfield

The "overwrite" scheme only needs loop through the input ports twice, but it comes with a large overhead on the tokens. The "loop" scheme needs to loop through the input ports k times (where k is the number of priority levels) with a small overhead on the tokens. There is a trade-off between the "overwrite" scheme and the "loop" scheme in terms of hardware complexity and speed requirement. Since the number of priority levels implemented is 2 in this development, the "loop" scheme is chosen to implement priority control for low overhead and easy implementation.

The CLP bit in a cell or the QOS bit in a SVP is used as a priority index. The number of priority levels is 2.

3.3.3 Buffer Management

The next issue is how to manage the buffer at each input port for packets with different priorities. The first one is to use the "threshold" scheme. In this scheme, there is only one queue and all the arriving packets are stored in the queue in a FCFS fashion. When a low-priority packet arrives, the number of low-priority packets in the queue is checked. If the number is above a certain threshold, the low-priority packet is dropped. The second one is a variation of the first one. In this scheme, there is only one queue and all the arriving packets are stored in the queue in a FCFS fashion. Low-priority packets are allowed to enter the queue only if the queue length is less than a certain threshold. The third one is to use the "push-out" scheme. There is only one queue and all the arriving packets are stored in the queue in a FCFS fashion. When the buffer is full, the new arrival high-priority packets will push the low-priority packets out of the queue (if there is any) and be stored. The fourth one is a variation of the third one. The high-priority packets are stored in front of the low priority packets. When the buffer is full, the new arrival high-priority packets will push the low-priority packets out of the queue and be stored. Within the same priority, the packets are stored in a FCFS fashion. The first three schemes are effective for an FPS with

output queueing, but not for an FPS with input queueing. The reason is that the HOL blocking for one class of packets interferes the other class of packets. Although the fourth scheme can largely improve the performance of the high-priority packets, the performance of the low-priority packet is uncontrollable. The fifth one is to use complete partitioning scheme. In this scheme, there is a separate queue for each priority. To completely eliminate the interactions among the packets of different priorities, the complete partitioning scheme should be used.

The complete sharing (one-queue) buffer management is not effective for priority control since segregation of connections based on their QOS cannot be performed. The best approach is to provide separate queues for each priority, i.e., using the complete partitioning buffer management. Since there are two priority levels, there are two queues in each input port, and each queue is used to hold the arriving packet based on their priority level. The insertion/removal of the packets to/from the queue should be implemented in a link list fashion. There is no upper limit for the high-priority queue length, but a limit is set for the low-priority queue length.

As discussed in Section 3.1, the throughput of a switch with input buffering is limited due to the head of line blocking problem. To increase the throughput of the switch, the input port has to examine more than one packets in the queue (or to use a larger "checking depth"). Since there are two queues in the input port, the checking depths for different queues with different priorities can use different values. Denote the checking depth for high priority queue d_h and that for low priority queue d_l. The packets in the high priority queue are examined before those in the low priority queue. It is assumed that the sum of the checking depths for two queues is fixed ($d_h + d_l = d$, where d is a constant), i.e, the maximum capacity of the switch is fixed. Now the issue is how to distribute the checking depths (or the switch capacity) to different services with different priorities to guarantee their QOS. The major concern with priority control is that although the packet loss ratio for the high priority packets can be guaranteed to be below a certain value (e.g 10^{-9}), the packet loss ratio for the low priority packets (e.g 10^{-6}) can not be guaranteed or can not be easily controlled. Using the centralized ring reservation scheme and different queues with different checking depths, it is found that the packet loss ratio for each priority level can be set to the desired value by adjusting the checking depth for each queue [3-10]. Furthermore, adjusting the QOS for the low-priority packets does not affect the QOS of the high-priority packets.

The maximum capacity for packets with high priority, denoted as $MAXC_h$, is achieved when $d_h = d-1$ and $d_l = 1$. The maximum capacity for packets with low priority, denoted as $MAXC_l$, is achieved when $d_h = 1$ and $d_l = d-1$. The values of $MAXC_h$ and $MAXC_l$ are fixed. By varying d_h and d_l (with $d_h + d_l = d$), the actual capacity of switch allocated for packets with different priorities is adaptable. The values of dh and dl should be programmable to account for different traffic scenarios. When the amount of high-priority traffic is small, the checking depth (d_h) for high-priority queue can be small and the checking depth (d_l) for low-priority queue should be large; and vice versa. The determination of the values (d_h and d_l) should be part of admission control procedure.

The reason that the values of $MAXC_h$ and $MAXC_l$ are fixed is because the high priority queue is always examined before the low priority queue. To make the values of $MAXC_h$ and $MAXC_l$ also adaptable, the high priority queue and low priority queue can be

examined in a prespecified order. For example, the sequence of examining the high priority queues first and then the low priority queues can be reversed once every m cycles. With this mechanism, the maximum capacity ($MAXC_h$) for high priority packets and that ($MAXC_l$) for low priority packets are adjustable. The maximum capacity for packets with different priorities is a function of M.

3.3.4 The Proposed Scheme

The recommended approach is to have a separate queue for each priority at each input port. If the number of priorities is large, use the "overwrite" scheme. If the number of priorities is small (≤4), use the "loop" scheme. Since the number of priority levels implemented is ≤ 4 in this development, the "loop" scheme is chosen for low overhead and easy implementation. The proposed multicast crossbar switch configuration considering two priorities is shown in Figure 3-16.

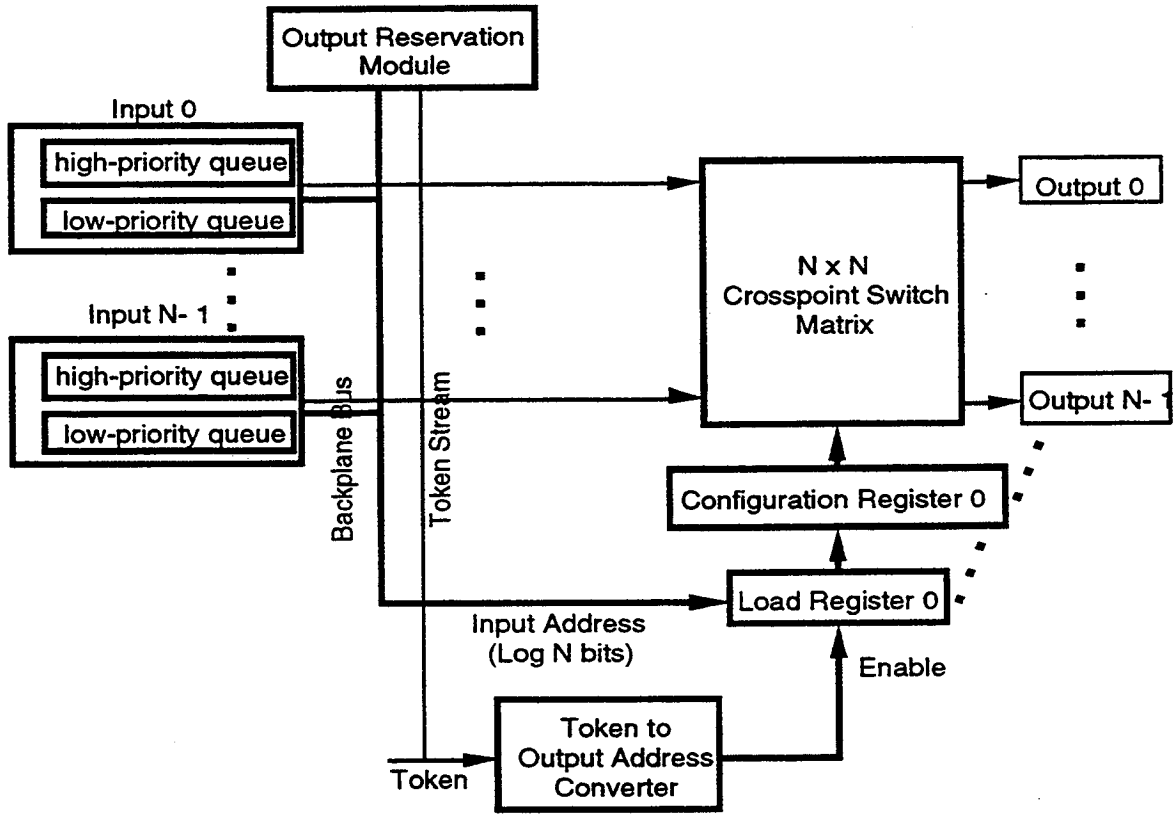


Figure 3-16; Multicast Crossbar Switch Configuration Considering Two Priorities

3.4 Integration of Circuit and Packet Switched Traffic

This subsection addresses the design considerations of integrating circuit and packet switching for a fast packet switch. The difference between circuit switching bandwidth assignment and packet switching bandwidth assignment is addressed first. The alternatives of implementing an integrated switch are discussed. Based on the discussion, one approach is recommended.

Conventional circuit switching employs deterministic multiplexing. All the required capacity from the source to the destination is reserved in advance. When a terminal receives a call request, the terminal sends this request to the scheduler. The scheduler assigns a number of uplink slots to the sending terminal and a number of downlink slots to the receiving terminal. The uplink and downlink access schemes use TDMA (same as ACTS). A channel is set up from the uplink slots to the downlink slots. (For the sake of discussion, only simplex call is established.) These allocated slots are used exclusively by the circuit connection. The slot position (in a TDMA frame) itself identifies the source and destination addresses. The on-board switch routes the connection based on the slot position. The switch path for a circuit connection is set up and reserved by the scheduler.

Packet switching employs statistical multiplexing. The terminal does not request capacity on a packet-by-packet basis. For B-ISDN, the required bandwidth for a new connection is based on the available information in the call setup message and the current network loading status. The bandwidth assignment procedures are part of admission control. Admission control is not considered in this study. Assume the required bandwidth (in terms of number of slots) for a connection is computed by the scheduler following a bandwidth assignment procedure. Assume the uplink access scheme uses TDMA and the downlink access scheme uses TDM. The scheduler assigns a number of uplink slots to the sending terminal and reserves a number of downlink slots. The packets destined to different destinations at the sending terminal can use any uplink slot assigned to the terminal for transmission. The destination address is included in the packet header. When the packets arrive to the on-board FPS, the FPS has to perform output contention resolution since packets from different input lines may be destined to the same output port at the same time. The design consideration of output contention resolution is addressed in Section 3.1. Some recommendations are made. After contention is resolved, the packet is self routed through the switching fabric. When the packet is routed to the proper output port, the packet can use any one of the reserved downlink slots for transmission.

There are two ways of providing both circuit switching and packet switching functions on-board. The first one is to use two separate switches: one circuit switch and one packet switch. The second one is to use one integrated fast packet switch. In this study, it is assumed an integrated FPS is used to provide service for both circuit switched and packet switched traffic. The basic requirements of an integrated switch are to provide services for both circuit switched and packet switched traffic and to preserve the QOSs of each connection.

An integrated switch has the following advantages. Integration simplifies the network management functions and the makes the introduction of new services with different

characteristics easier. Integration also provides simpler implementation and control, less hardware, easy fault tolerance structures, reduced mass and power, and unified routing procedure. Most importantly, the integrated switch is more flexible in allocating the capacity of the switch between circuit and packet switched traffic.

The circuit traffic is segmented into packet formats (such as cells or SVPs) at the sending terminals and reassembled into channel formats at the receiving terminals. Both packet and circuit have the same packet format. The uplink uses the slotted transmission format (TDMA). The unified packet format occupies one slot of uplink frame. The integrated access scheme uses a combination of TDMA and packet transmission.

In general, there are two approaches to emulate the circuit switching operation using a fast packet switch. The first one is to reserve the switch path for circuit packets belonging to the same connection each frame. Since the switch path is reserved, the circuit packets are guaranteed to pass through the FPS without any queueing delay. There is no output contention among circuit switched traffic. The output contention between circuit switched traffic and packet switched traffic can be eliminated using priority control. Consequently, delay jitter of circuit connections is minimized. The second one is to reserve the switch capacity for circuit packets each frame. Since only switch capacity is reserved, circuit packets may still have output contention with other packets. Circuit packets may suffer (a small amount of) queueing delay at the switch. A smoothing buffer is required at the receiving terminal to compensate the delay jitter. Designate the first approach "Switch Path Reservation Scheme" and the second approach "Switch Capacity Reservation Scheme". Both approaches are discussed.

3.4.1 Switch Path Reservation Scheme

In this scheme, the scheduler not only assigns slots to the sending terminals for a circuit connection request, but it also resolves switch output contention for circuit slots (connections) belonging to different uplink TDMA carriers. (This is similar to ACTS scheduling or SS-TDMA scheduling.) The capacity scheduling and switch output contention can be performed using a centralized scheme or a distributed scheme.

Based on a previous study for SPAR, centralized capacity scheduling is more advantageous than distributed capacity scheduling comparing signaling capacity, scheduling assignment conflicts, scheduling assignment delay, information synchronization, scheduling algorithm flexibility, reliability, and processing requirement. Nevertheless, both centralized and distributed schemes are discussed.

A centralized scheme is discussed first. Before a slot can be assigned, the scheduling algorithm has to examine several constraints such as the maximum capacity of the sending terminal, the maximum capacity of the receiving terminal, and the total system capacity. In order to reserve the switch path, switch output contention resolution for circuit switched traffic becomes another constraint from the view point of the scheduling algorithm. After the uplink slots are assigned to the terminal, the terminal uses the slots for transmission. Designate the circuit switched packets as circuit emulation (CE) packets. Since the (CE) packets are sent from the terminals at specified slots, they will also arrive to the switch at specified time every frame. Remember the switch path has been reserved for the CE

packets (by the scheduler). Assign CE packets with high priority and data packets with low priority. Let the CE packets participate in the output contention resolution with other data packets at the switch. Since CE packets have a higher priority than the data packets and the scheduler already resolves output contention among CE packets at the same slot time, transmission of the CE packets at the next slot time is guaranteed. Since the CE packet is prepended with a routing tag, it can be self routed through the switching fabric. No circuit switched buffer is required at the input port in this case.

A distributed scheme is discussed as follows. Each terminal has a complete busy/idle status of the slots in the system. Each busy circuit slot is associated with one destination (or multiple destinations for multicast connections). The terminal assigns slots to itself after the capacity scheduling and output contention resolution are performed locally. The terminal sends out the assignment information to all the terminals for confirmation. After two round trip delay, if the terminal gets an assignment conflict message, the terminal schedules another set of slots. Otherwise, the terminal can start transmission. The output contention resolution scheme for CE packets at the switch follows the discussion in the previous paragraph.

3.4.2 Switch Capacity Reservation Scheme

In this case, the CE packets (belonging to different connections) can be transmitted at any assigned slots at the sending terminal in one frame. This implies that the capacity at the terminal is shared by multiple circuit switched connections and possibly by packet switched traffic. The circuit switched traffic is transmitted on these assigned slots before the packet switched traffic. If there is capacity left and there is packet switched traffic, then packet switched traffic can use the remaining capacity for transmission. This increases the bandwidth efficiency due to statistically multiplexing. Since arrivals of CE packets to the on-board switch are not periodical, there is no need to reserve the switch path for the CE packets every frame. The scheduler is only responsible of allocating slots without resolving output contention among CE packets. However, priority control is required to bound the circuit switched delay jitter at the switch. The priority control with the centralized ring reservation scheme has been discussed in Section 3.3. Some necessary modifications are described below. The priority of a CE packet is higher than that of a data packet during output contention resolution. If two CE packets are destined to the same output port at the same time, one of the CE packets is chosen randomly. When a CE packet loses the contention resolution, the CE packet becomes an old CE packet. The priority of an old CE packet is higher than that of a new CE packet. This implies that the number of priorities is at least four by considering old CE packets, new CE packets, high-priority data packets, and low-priority data packets. A small circuit switched buffer is required at the input port to hold the CE packets if output contention among CE packets does occur. It is expected the size of the circuit switched buffer is very small.

3.4.3 The Proposed Integration Scheme

The switch path reservation scheme places a high processing requirement on the scheduler, and there is no statistical multiplexing gain for circuit switched traffic at the terminal and the switch. However, the circuit switched traffic delay jitter is minimized. The switch capacity reservation scheme has less demand on the scheduling processing

requirement, and the terminal capacity and the switch capacity are fully utilized. The circuit switched traffic suffers queueing delay at the terminal and the switch. The delay jitter has to be compensated at the receiving terminal.

The switch capacity reservation scheme is recommended. There are three logical subqueues at the input port, one for the circuit switched traffic, one for high priority packets and one for low priority packets. The insertion/removal of the packets to/from the subqueue should be implemented in a link list fashion. There is no upper limit for the circuit switched data queue length and the high-priority queue length, but a limit is set for the low-priority queue length. The proposed multicast crossbar switch configuration considering integrated operation is shown in Figure 3-17.

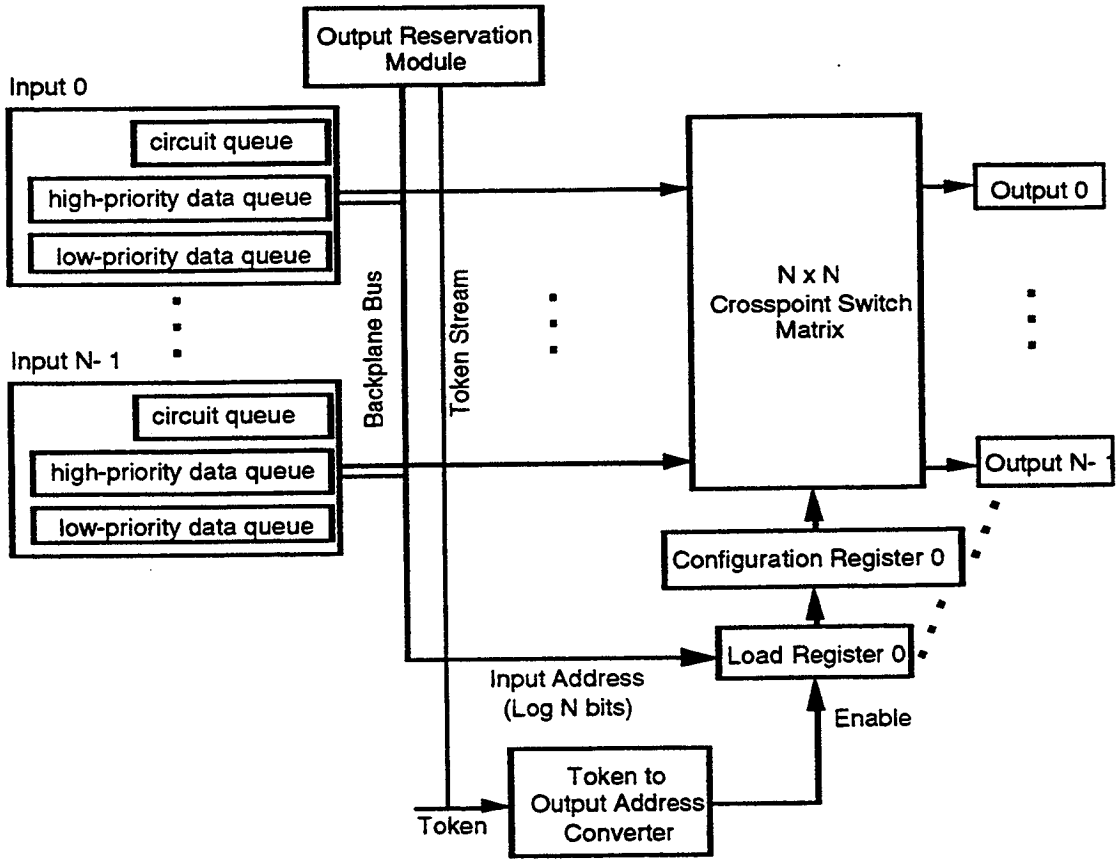


Figure 3-17: Multicast Crossbar Switch Configuration Considering Integrated Operation

3.5 Fault-Tolerant Operation

A system with fault-tolerance is defined as a system which is able to perform the assigned functions correctly in the presence of either hardware failures or software errors. Without a proper fault-tolerant design, a single point failure in the FPS may cause the whole satellite communications system unoperational. Conventionally fault-tolerance can be achieved using two approaches: hardware redundancy and software coding. The degree of fault tolerance depends on how well the faults can be detected and repaired or replaced. Clearly the more hardware/software redundancy is put into the system, the system has a higher fault tolerance. However, more redundancy also means higher cost, mass, and power. The procedures used to detect the faults, locate the faults, and reconfigure the system to be fault free are overhead functions; they also may create a new class of faults which the basic system does not have. A fault-tolerant design of the FPS must be started at the very earliest stage such that a trade-off among the additional cost, mass and power as a result of fault-tolerant design, reliability and performance can be made.

The first step to achieve fault tolerance is to eliminate hardware design and software development faults. The general requirements to design a fault-tolerant FPS were reported in Reference 3-11. They are:

- the probability that packets are duplicated, dropped or corrupted due to a fault should be minimized
- a failure of components will not affect the existing connections
- corrupted ATM cells (as a result of faulty component) will not affect other normal cells
- The OBC should be able to detect any faulty path in the switch
- All the modules (including the redundancy) must be checked for health status by the OBC
- testing of different modules should not interfere the existing traffic
- There should be no single point failure.

Another alternative to check the FPS modules is to use ground terminals. Both fault detection and fault diagnosis procedure executed by the OBC and ground terminals are presented in this subsection.

For on-board FPS, the other requirement is that the power consumed by the redundant units should be minimized. In general, the on-board redundant units are cold standby. Some vital components must be duplicated such as clock, power supply, and OBC.

A module should be designed in such a way that any single point failure should not disable the module. For example, failure of the power pin disables the whole chip if the chip only has one power pin. Therefore, the chips used in the FPS should have spare power pins. There are two approaches of configuring the spare power pins. Let the total

number of power pins be p. The first approach is that each power pin handles $\frac{1}{p}$ of power loading. If one power pin fails, each power pin handles $\frac{1}{p-1}$ of power loading. The second approach is that the spare power pins are in open circuit mode. When the master power pin fails, one of the spare power pins is switched back on. The power pin and the redundant pins may come from different power sources. It may not be justifiable to duplicate the whole module simply because there is only one power pin in the module. Instead of duplicating the module, more power pins should be provided. Evidently if off-the-shell components are used in building the FPS, the internal design of the components can not be modified. If a component is built in house, special fault tolerance can be added into the design (such as spare power pins).

In this subsection, fault tolerant design is discussed only at the system level. The component fault tolerant design (such as memory chips and I/O units) and internal fault tolerant design (such as memory cells) will be exploited in the "High Level Design" task. General fault tolerant techniques are discussed; only part of the techniques will be used in the breadboard design., The hardware redundancy approach is discussed for OBC, input ports, switching fabric, output ports, and output reservation module. The software redundancy approach (coding) is discussed for memory. In general, fault tolerant operation comprises the following steps [3-12]:

- fault detection: to determine whether a fault exists using hardware or software approaches.
- fault diagnosis (fault location): to pinpoint a fault using hardware or software approaches.
- fault isolation: to prevent the errorneous information from propagating to other healthy components.
- fault replacement/reconfiguration: reconfigure the system such that the faulty module is replaced or repaired.

3.5.1 OBC Fault Tolerant Design

The OBC is the most crucial component for FPS fault-tolerant operation. The high-level functional description of an OBC is described in Section 4. OBC must have at least 1-for-1 redundancy. Furthermore, OBC may have two input ports and two output ports connected with the switching fabric to facilitate fault detection procedure and to increase fault tolerance. The OBC must have self-diagnostic software and self-checking circuit to detect its own errors and faults.

There are three different fault-tolerant architectures for the OBC with 1-for-1 redundancy [3-12].

In the first architecture, both controllers perform self-diagnosis (or external-diagnosis by the ground station). One OBC is the master and the other one is the slave. If the master fails, the slave is notified by some detection circuit. Take the communication controller developed at E-Systems Inc. as an example [3-13]. The master is required to reset a timer at every fixed interval. If the timer is not reset, a (separate) time-out circuit notifies the slave

and the slave resumes the operation. The disadvantage of this architecture is that if the time-out circuit fails to notify the slave controller, the system is stuck with the faulty master controller.

In the second architecture, both controllers perform self-diagnosis (or external-diagnosis by the ground station) and also cross-diagnosis. There are a primary controller and a secondary controller. The cross-diagnosis is to check and test the status of the other controller. Both controllers are required to perform a certain operation such as store the status of some registers in a shared memory. Both controllers are required to place data (health status) into the shared memory, and both of them also check the memory contents constantly. If one of the controller fails to place the data in the memory, an indication of failure in the controller occurs. The error-free controller resumes the operation. If one controller is in fault, the system becomes a one-controller system. In this case, the remaining controller keeps performing self-diagnosis. If the controller detects errors and it can not recover the errors, the controller should notify the ground control station. The ground control station may disable the controller and control the FPS directly from the ground.

In the third architecture, not only both controllers perform self-diagnosis (or external diagnosis by the ground station) and cross-diagnosis, but also both controllers compare the results of each control action (replication check). If the results do not match, faults occur at one of the two controllers. If the self-diagnosis and cross-diagnosis are able to identify the faulty controller, the remaining controller is the primary controller. If both controllers considered themselves to be fully operational, then the ground control station has to perform external diagnosis to identify the faulty controller.

The other alternative is to use the majority voting architecture. This approach was adopted in Reference 3-14 to design an on-board satellite switch controller for microwave switch matrix. There are four redundant controllers. In normal operation, three controllers are operated with 2 out of 3 voting and the other one is cold standby. If one of the three controllers fails, the cold standby becomes operational. If two controllers fail, the remaining two controllers performs replication check for their results (the third architecture mentioned above). The control data stored in the controller is FEC encoded. The control data is decoded when it is read out from the controller. When the results from the remaining two controllers do not match, the result which does not have any error is selected. After this, the system becomes a one-controller system.

External diagnosis of the OBC from a ground station to verify its operation is always required.

The OBC communicates with different input ports and output ports through the switching fabric. To improve the fault tolerance, the OBC may have two inputs (input N and N+1) and two outputs (outputs N and N+1) connected to the switching fabric (see Figure 3-18). An input port can use either output N or output N+1 to communicate with the OBC. If the OBC is capable of processing the incoming packets twice faster than the link speed, the OBC can receive two packets simultaneously from different input ports. If not, the OBC can only receive one packet at a time. The extra input and output is provided only for redundancy.

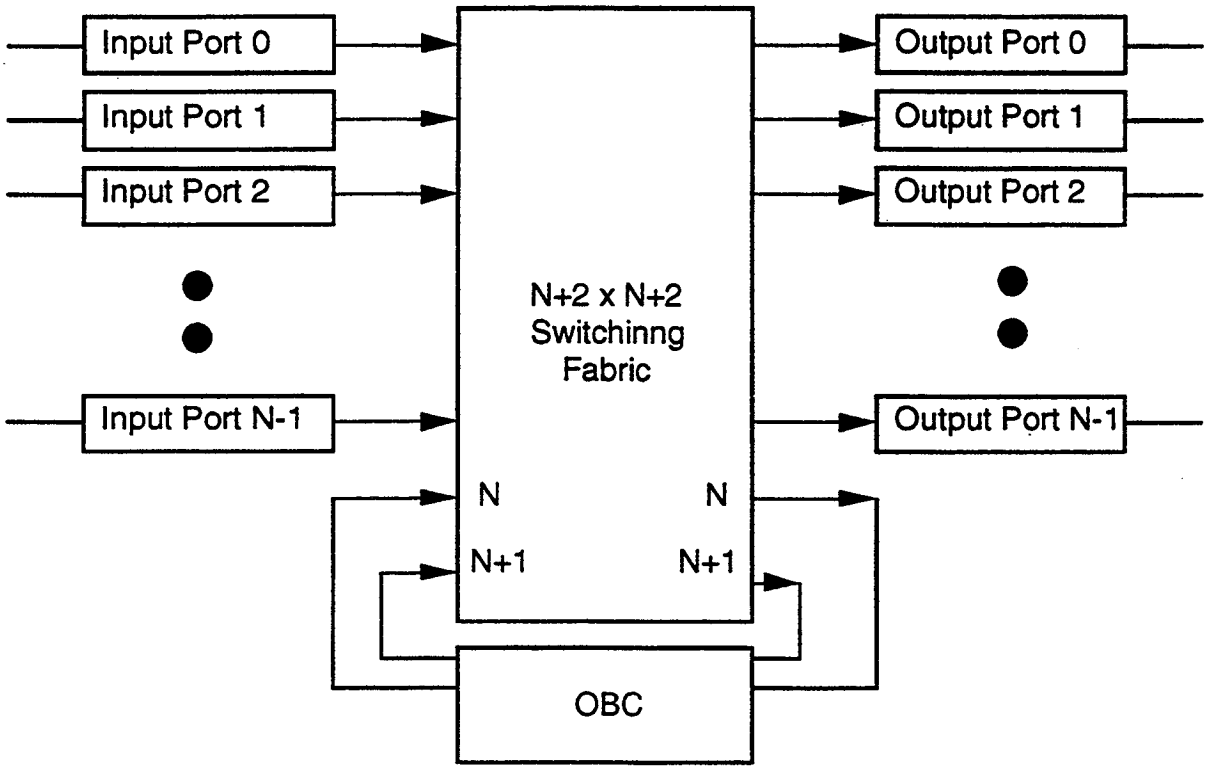


Figure 3-18: Basic Configuration of FPS

Basically, the OBC is a microprocessor-based module. The fault detection and diagnosis procedure for a microprocessor-based module was discussed in Reference 3-13. The following discussion follows that in Reference 3-13.

The fault detection procedure is to apply a set of input test patterns to a component and the output responses are verified. If the verification fails, faults occur at the component under test.

There are two testing approaches: concurrent and explicit. In the concurrent approach, the data itself (to be processed by the component) is the test pattern. A monitoring circuit examines the output response. A typical example is the parity checking (error detection coding) or error correction coding. Test with coding can be applied to RAMs and LSI circuits. These devices are referred as self-checking circuits since a fault can be detected by verifying the output of the component [3-15]. The self-checking circuit may create an additional cost of 20% compared with the original circuit [3-12].

In the explicit approach, special data is used as test patterns. The test can be performed externally or internally by the component itself. The test patterns can be stored in a ROM or be computed in real time using an algorithm. A typical implementation of generating test patterns by an algorithm is to use a counter.

3.5.2 Output Port Redundancy

Two ways of implementing output port redundancy are identified.

3.5.2.1 1-for-N Redundancy

In 1-for-N redundancy, there is one redundant output port. The configuration is shown in Figure 3-19.

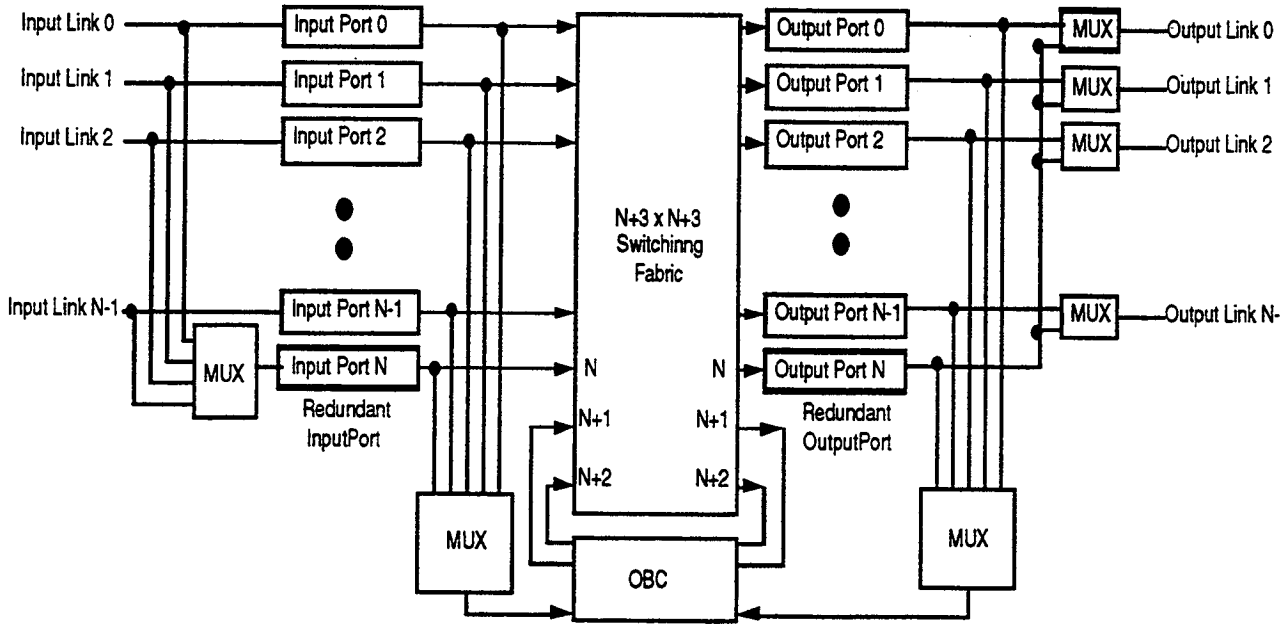


Figure 3-19: 1-for-N Redundancy Configuration for Output Port

3.5.2.1.1 Fault Detection by OBC

Either on-line testing or off-line testing can be adopted. Testing should be performed when the traffic loading is light. For off-line testing, the output port N can replace the output port under test temporary and isolate output port under test completely.

A loop-back method is used to check the output ports (and the input ports). Let output port 0 be the module under test. The OBC sends packets with known patterns using input N+1 through the switching fabric to output port 0. Output port 0 sends the packets through the multiplexer back to the OBC. Sending these test packets to output port 0 may slightly increase the switch loading. When the OBC receives the test packets, the OBC check the patterns to see if they match with the original ones. If they do not, faults are detected and fault-tolerance operation enters fault diagnosis stage. If no faults are found, the OBC tests the next output port following the same procedure.

The redundant output port N should also be checked by the OBC constantly to make sure it is at the good state.

3.5.2.1.2 Fault Diagnosis by OBC

If faults are detected, faults may occur at the output port 0 or the switching fabric (between input N+1 and output 0). The OBC sends the same test packet through a different switching fabric path (using input port N+2) to output port 0. If faults still exist, output port 0 are faulty; otherwise, one of the switching fabric paths is in error.

3.5.2.1.3 Fault Recovery and Fault Reconfiguration by OBC

When faults are found at the output port, the OBC has to determine whether the faults can be recovered. For example, a memory error can be recovered by instructing the read/write operation to skip the errorneous locations. If the faults are not recoverable, the output port will be isolated and be put in off line. The multiplexer is set in such a way that output line 0 will receive traffic from output port N (see Figure 3-19). If the routing tag of the packet is prepended at the input port, the routing tag translation tables have to be updated at each input port to reflect the replacement. If the routing tag is added at the ground terminals, then the OBC has to send special instructions to all the terminals to update the translation tables.

3.5.2.2 m-for-N Redundancy

There is a (N+m) x N output relay switch in front of the N+m output ports. The switch size is (N+2+m) x (N+2+m). There are m redundant output ports available. This configuration is shown in Figure 3-20. The fault detection and fault isolation can apply the same procedure mentioned above. When faults are detected in one of the N output ports, the defected output port is put off-line and the defected output port is replaced by one of the redundant output ports. The switching state of the output relay switch must be changed to reflect the new configuration. The OBC may have to update the translation tables at each input port if the routing tag is appended at the FPS to reflect the new configuration. If the routing tag is added at the terminals, then the OBC has to send special instructions to all the terminals to update the translation tables.

3.5.3 Input Port Redundancy

After the output ports have been tested to be error free, the input ports are tested. Two ways of implementing fault tolerance at the input ports are identified.

3.5.3.1 1-for-N Redundancy

The switch size used is (N+3) x (N+3). Input port N is provided as a redundancy input port. An N x 1 multiplexer is used to connect N input lines to the extra input port N. The configuration of 1-for-N redundancy for output port is shown in Figure 3-19. When an input port is detected in fault by the OBC, a fault-reconfiguration algorithm will replace the faulty

input port with the input port N such that the switch operation can be continued with minimal disruption.

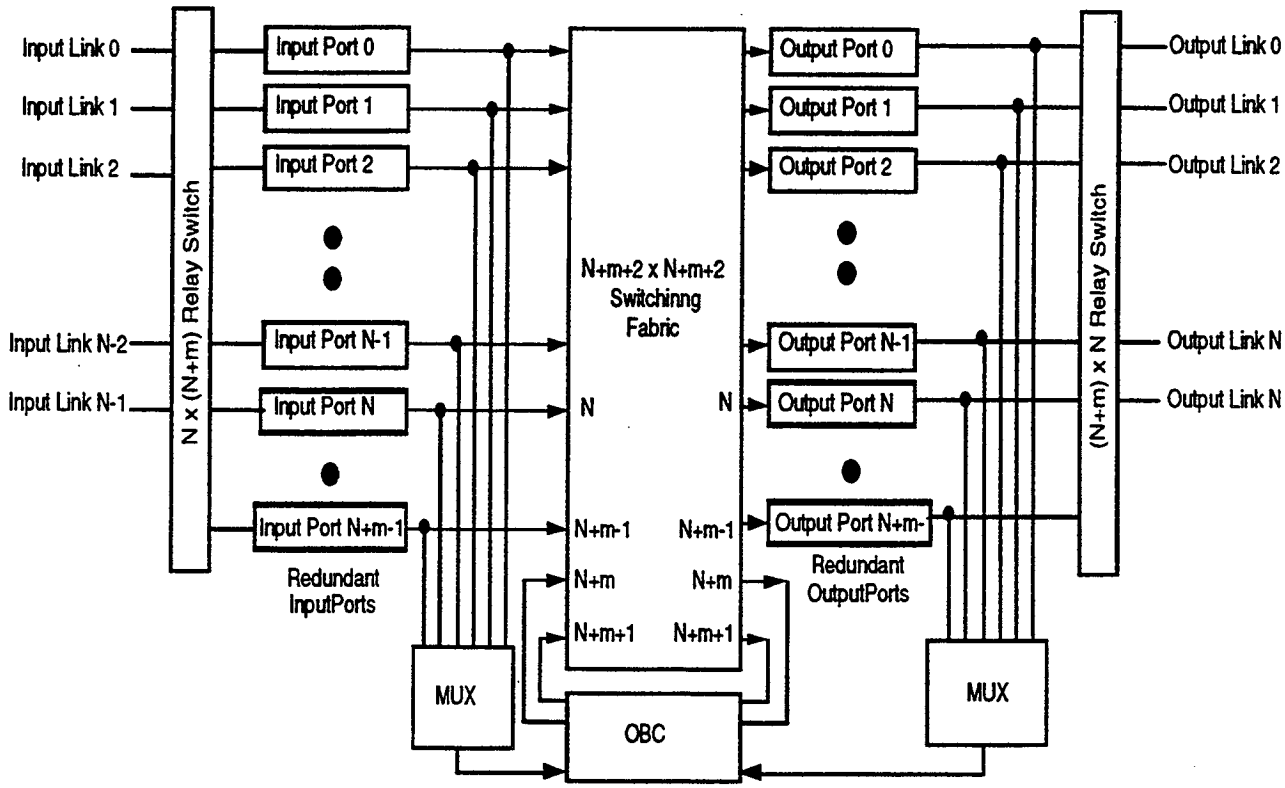


Figure 3-20: m-for-N Redundancy Configuration for Output Port

3.5.3.1.1 Fault Detection by OBC

The fault detection procedure is to check the health status of each input port. Either on-line testing or off-line testing can be adopted. Testing should be performed when the traffic loading is light. For off-line testing, the input port N can replace the input port under test temporary and isolate input port under test completely.

A loop-back method is used to check the input ports (and the output ports). Without loss of generality, let input port 0 be the input port under test. The OBC generates the test packets and send these packets to output port 0. When output port 0 receives the test packets, output port 0 relays these packets to the corresponding input port 0. Input port 0 sends packets back to OBC through the multiplexer.

When the test packets arrive to OBC, a comparison circuit is used to compare the contents of the packets with the patterns stored in the OBC. If faults are detected, the fault-tolerance operation enters the fault diagnosis stage. If no faults are detected, the OBC test the next input port (input port 1) following the same procedure.

The redundant input port N should be checked by the OBC constantly to make sure it is at the good state.

3.5.3.1.2 Fault Diagnosis by OBC

When faults are detected, fault diagnosis is required to pin point the faulty component. If faults are existed when the input port 0 is tested, the faults may occur at the switching path (between input N+1 and output 0), the output port 0 and the input port 0. Since the output ports are tested before the input ports, the output ports can be assumed to be fault free. The OBC sends the test patterns to output port 0 through input N+2. If faults still exist, the faults must occur at input port 0.

3.5.3.1.3 Fault Recovery and Fault Reconfiguration by OBC

When faults are found at the input port, the OBC has to determine whether the faults can be recovered. For example, a memory error can be recovered by instructing the read/write operation to skip the erroneous locations. If the faults are not recoverable, the input port will be isolated and be put off line. The multiplexer is set in such a way that input line 0 uses input port N for storage.

3.5.3.2 m-for-N Redundancy

There is a N x (N+m) input relay switch in front of the N+2+m input ports. The switch size is (N+2+m) x (N+2+m). There are m redundant input ports. The redundancy configuration is shown in Figure 3-20. When faults are detected in one of the N input ports, the relay switch will bypass the defected input port and switch the traffic into another input port. The OBC may have to download the translation table to the new adopted input port if the routing tag is appended at the OBC. The fault detection and fault diagnosis can apply the same procedure mentioned above.

3.5.4 Switching Fabric Path Redundancy

The switching fabric redundancy can be achieved by increasing the number of paths in the switching fabric.

3.5.4.1 Switching Fabric 1-for-N Redundancy

3.5.4.1.1 Fault Detection by OBC

There are N^2 crosspoints in the crossbar switch. Each of these crosspoints has to be tested. The OBC sends test packets to output port 0 first. Output port 0 relays the test packets to the corresponding input port 0. Input port 0 sends these test packets to different output ports sequentially. When an output port receives the test packets, the output port sends the test packets through the multiplexer to the OBC. The OBC examines the test packet contents for error detection. This procedure is repeated for every input-output pair.

3.5.4.1.2 Fault Diagnosis by OBC

The fault-detection procedure mentioned above has achieved fault diagnosis naturally. When errors are found in the test packet, the faulty switching path is the input-output pair under test.

3.5.4.1.3 Fault Reconfiguration by OBC

After the faulty path (crosspoint) has been identified, the faulty path is isolated. Assume the switching path between input 0 and output 0 is faulty. The packets, destined to output port 0, at input port 0 are routed to output port N using the redundant switching path. Since output port N is not the final destination, the output port N sends these packets to output line 0 by enabling the associated multiplexer.

3.5.4.2 Switching Fabric 1-for-1 Redundancy

If the switch chip does not have any redundancy, a failure of the power pin or the ground pin causes a failure of the whole system. A 1-for-1 redundancy for the switch chip is required. These two switch chips may receive power from two different power sources. The other option is that to design a chip with spare power pins and spare ground pins. The power pins may receive power from different power sources. By providing both switching fabric redundancy and switching fabric path redundancy, the fault-tolerant operation becomes very robust. Assume the switching path between input 0 and output 0 of the switching fabric 1 is faulty. Then the OBC can instruct input port 0 to send packets, destined to output port 0, via switching fabric 2.

3.5.5 Fault Detection, Fault Diagnosis, and Fault Reconfiguration by Ground Terminals

When OBC fails to perform fault tolerant operation, it is desirable that the ground terminals can take over the operation. It is required that each beam has a terminal to participate the testing. Let the terminal located in beam i denoted as terminal i, where $0 \leq i \leq N-1$. Since any two disjoint paths in the FPS can identify a point (a module), a pair of terminals are required to test one module. The module can be input port, switching path, or output port. Let output port 0 be the module under test. Terminal 0 sends the test packet to itself. If faults occur, either input port 0, the switching path between input 0 and output 0, or output port 0 is faulty. Terminal 1 sends the test packet to terminal 0. If faults still exist, output port 0 is identified to be faulty. If faults do not exist, either input port 0 or the switching path between input 0 and output 0 is faulty. Terminal 0 sends the test packet to Terminal 1. If faults still exist, input port 0 is identified to be faulty. If faults do not exist, the switching path between input 0 and output 0 is identified to be faulty. The same procedure is followed to test the other modules. The fault reconfiguration procedure is executed when commands from the ground terminal are received. The disadvantage of testing by ground terminals is that the test can not be executed in real time, since the received test data from different terminals must be processed at a central location.

3.5.6 Output Reservation Module Redundancy

A fault tolerant centralized input ring reservation scheme is vital to the FPS. If a (unrecoverable) failure in the centralized input ring reservation module occurs, the whole FPS is unoperational. The input ring must have a 1-to-1 redundancy. As shown in Figure 3-21, there are two reservation modules; one at the top of the input ports and one at the bottom. These two reservation modules are connected by two backplane paths (buses). Since the path passes through every input port, a failure of the input port disconnects the path. The path through the input port should have a bypass mechanism such that a faulty input port can be isolated from the path. After the top module sends the tokens through the input ports, the bottom module processes the tokens. If an error is detected in the tokens, a failure of the path is detected. The OBC should perform fault diagnosis and pinpoint the faulty segment. If the fault can not be recovered, the bottom module will resume the operation and the top module checks the status of tokens. The bus used to connect the input ports can be designed to have spare lines. If one line fails, the spare line can be put in operation.

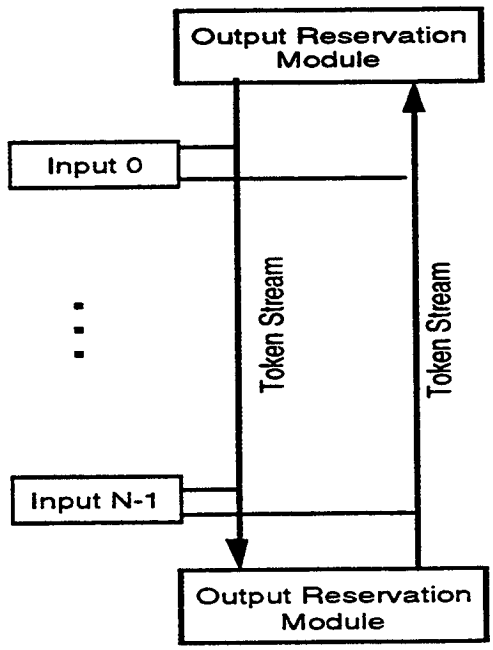


Figure 3-21: Redundancy Configuration for the Output Reservation Module

3.5.7 On-Board Control Memory Coding for Soft Failures

There are four effective ways of providing fault-tolerance for the control memory and data memory [3-16]. The first way is to use a error detection code (even/odd parity check) along with the information. The information is encoded before being stored in the memory. When the information is read out from the memory, if error is detected, a diagnosis procedure is invoked and that particular memory location is examined. If faults are found

in the location, the OBC will send out commands to skip the faulty location. The second way is also to use only a error-detection code. After memory fault-diagnosis procedure is applied, the faulty memory location can be identified. Hardware correction is possible using a register in the decoder. The bits in the faulty positions can be corrected using bit-operation. The third way is to use an error correction code along with the information. The larger the correction capability, the more faults can be tolerated; however, more redundant bits are used, and more complexity and more delay are required for the decoder. The fourth way is to use the majority voting technique.

The distribution control unit (DCU) in Intelsat VI uses three identical (control) memory modules. The DCU is responsible for setting up the connection states of the microwave switch matrix. Switch configurations are stored in the DCU memory. One memory is on-line while the other two are used for stand-by and off-line function. The off-line memory module has the same data contents as the on-line memory module. Once failures are found from the on-line module, fast switchover can be made. The data are encoded from the ground station using the Hamming code before it is stored in the memory. When the data is read out from the memory, forward error correction is applied to the data. The chosen Hamming code has one-bit error correction capability. The memory contents are refreshed immediately after they are read out. That is to say after the data is read out and decoded, the data is re-encoded and stored at the same memory location. This is to prevent accumulation of errors in the same (control) memory location.

The off-line memory can be read out and sent to the ground station for error verification/correction to make sure that the memory module is always in the good state.

The control memory in Reference 3-14 is protected using the combination of error correction code and majority voting.

In addition to providing protection for the memory, the read/write logic for the memory must be doubled.

High-level functional requirements for each module of the fast packet switch are described below.

4.1 Input Port

The input port accepts the demodulated satellite virtual packets (SVPs). It performs synchronization to identify the packet boundary. The SVP is stored in the proper location of the proper queue based on the SVP traffic type and loss priority. Output contention resolution is performed for every packet in the queue(s). After output contention is resolved, the SVP is then ready to be sent to the switching fabric.

The input port high-level functional block diagram is shown in Figure 4-1. The functions performed by each block are presented below.

The packet synchronization/header error control block has two major functions. First, it accepts the demodulated SVP bit stream and performs synchronization to identify the packet boundary. Second, it performs forward error correction for the SVP header. If errors are found and they are not correctable, the erroneous packets are dropped. If no errors are found, the SVP is ready to be stored in the buffer. If errors are found and they are correctable, the errors will be corrected and then the packet is ready to be stored in the buffer. Idle SVPs should be discarded. Idle SVPs can be identified by examining the VCN field in the SVP header. (For example, the first bit in the VCN field can be designated as an activation/inactivation bit.)

The traffic monitoring/congestion control block measures the switch loading (such as arrival rate, queue length, or link utilization). The measurement results are used by the OBC at regular intervals. When the switch is congested, the low-priority packets are dropped before the high-priority packets. (In real operation, the traffic monitoring device should also count the number of lost packets and the number of erroneous packets.)

The buffer block with its associated controller inserts the arrival SVP to the proper location of the proper queue based on the payload type and the loss priority.

The output port reservation with priority control block performs output contention resolution for every SVP in the buffer. High-priority packets win the output contention resolution when they are contended with low-priority packets. After the output port for a SVP has been reserved, the SVP is ready to be transferred to the switching fabric.

To facilitate testing, the input port is able to receive (test) packets from the corresponding output port. (For example, input port 0 can receive packets from output port 0.) The (test) packets are treated as a regular packet.

(Note that the routing tag of the SVP is added at the ground terminal, not at the input port.)

The functions of the transmission interface and switching fabric interface will be discussed in the "High Level Design" task.

Section 4

Switching Subsystem Functional Requirements

This section specifies high-level functional requirements for the on-board baseband switching subsystem which will be implemented in the subsequent tasks. The functional requirements are based on the analyses studied in Section 3 "Design Considerations for Switching Subsystem". The performance requirements are highly dependent on the services the switch provides. A general guideline for performance requirements is provided at the end of this section for delay-sensitive circuit traffic and loss-sensitive data traffic.

An on-board fast packet switch (FPS) consists of input ports, switching fabric, output ports, output port reservation module, and on-board controller (OBC). The high-level functional block diagram of the FPS is shown in Figure 4-1.

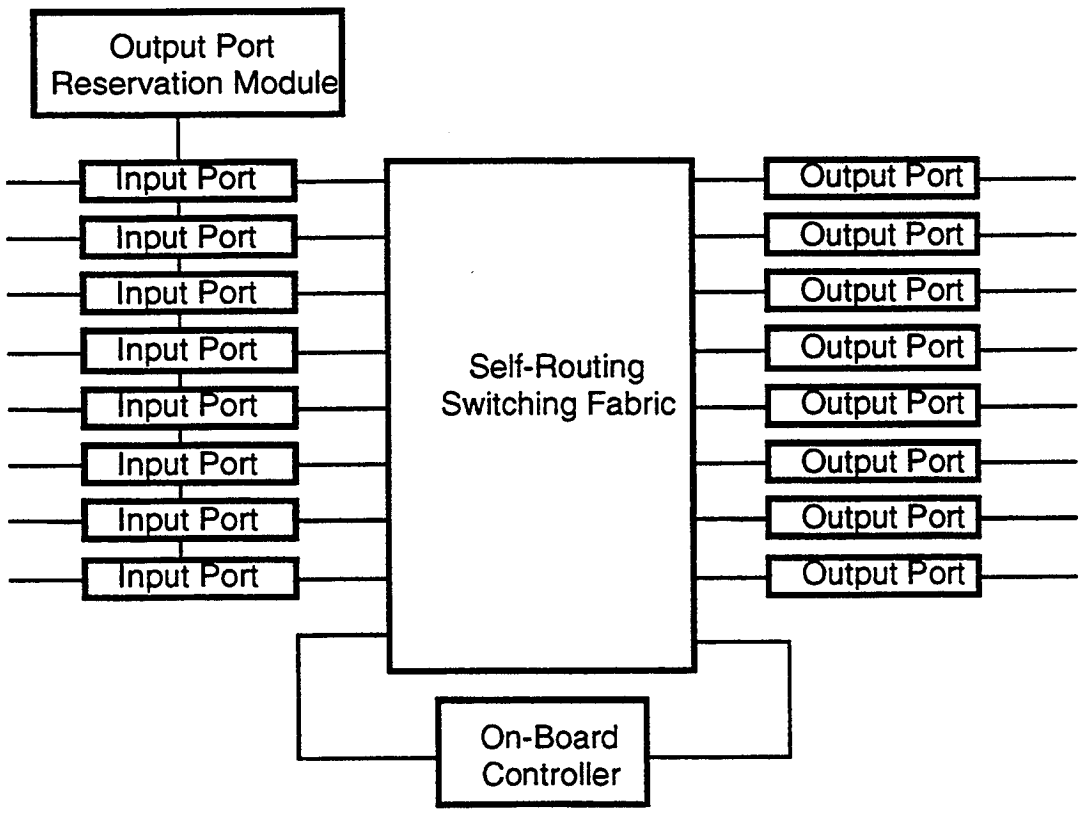


Figure 4-1: High-Level Functional Block Diagram of FPS Modules

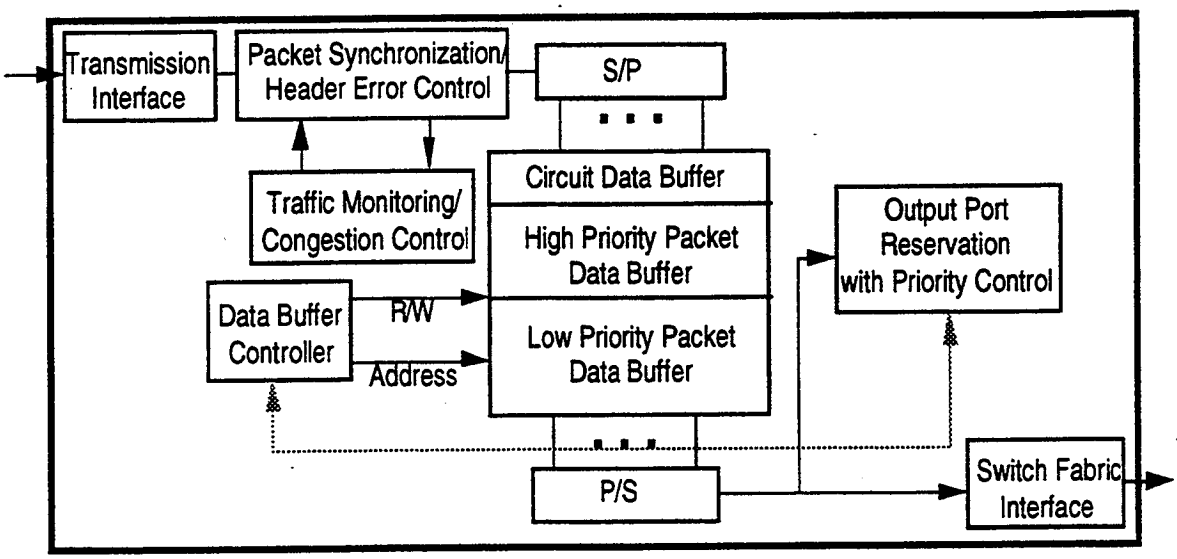


Figure 4-2: High-Level Functional Block Diagram of Input Port

4.2 Switching Fabric

The switching fabric accepts the SVPs from different input ports and routes these packets to the destinations solely based on the routing tag in the SVP header.

The switching fabric performs the following high level functions:

- self-routing: The switch fabric routes the packets solely based on their routing tags.
- multicast: The switch fabric routes a multicast packet to multiple destinations based on the packet's multicast routing tag.
- nonblocking: When the arrival packets have distinct routing tags, the switch fabric is able to route these packets to the destinations without any blocking.
- packet sequence preserving: The switching fabric will not transmit packets with the same VCN out of sequence.

(Note the chosen commercial crossbar switch does not have the self-routing capability. A special controller must be designed to control the crossbar switch such that the crossbar switch performs as a self-routing switch. The implementation of the special controller is described in the "High Level Design" task.)

C-2

4.3 Output Port

The output port accepts the SVPs from the switching fabric and performs speed and/or format conversion for downlink transmission. The high-level functional block diagram for an output port is shown in Figure 4-3. The functions of each block are addressed as follows.

The traffic monitoring block is to measure the switch loading (such as mean utilization). The switch loading results are used by the OBC at regular intervals.

The speed/format conversion block uses a FIFO to convert the speed and/or format suitable for downlink transmission. (One typical example is to convert uplink TDMA format into downlink TDM format.) To maintain downlink synchronization, when there is no packet to transmit, idle packets must be generated and inserted into the downlink bit stream.

The routing tag of the packet may be deleted at the output port.

To facilitate testing, the output port is able to relay the (test) packet to the corresponding input port. The test packet is also inserted into the downlink bit stream.

The functions of the transmission interface and switching fabric interface will be discussed in the "High Level Design" task.

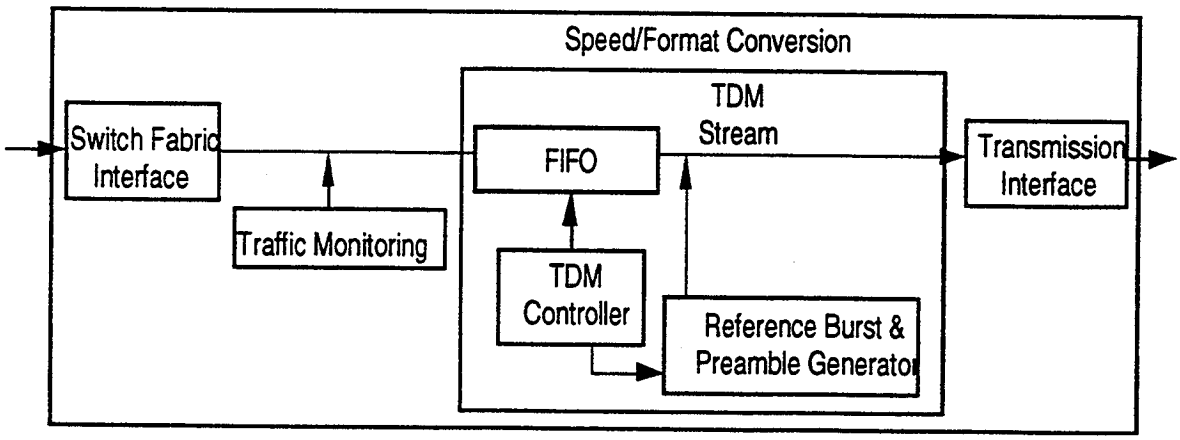


Figure 4-3: High-Level Functional Block Diagram of Output Port

4.4 OBC

The OBC accepts control and test packets from the input ports via the switching fabric. The OBC also accepts test packets via internal interconnection mechanisms from different ports. The OBC sends control and test packets to different output ports via the switching fabric. The OBC handles the following algorithms: congestion control and fault-tolerant operation. The OBC has knowledge about the switching network topology, the

functionalities of different modules (basic and redundant modules), and the interconnection mechanisms among different modules. The OBC monitors and collects switch loading at regular intervals. The switch loading is an input argument to the congestion control algorithm. An output of the congestion control algorithm is the amount of traffic increase or traffic reduction for the ground terminals. The OBC generates control SVPs containing traffic reduction/increase messages at regular intervals and send these packets to the ground terminals. The OBC sends the test SVPs to different ports via switching fabric when the traffic loading is light. When the test SVPs are received from the ports under test, the OBC executes the fault detection procedure. If faults are detected, the fault-tolerant operation enters the fault diagnosis stage. After the faulty module is pinpointed, the OBC replaces the faulty module with a redundant module by executing the fault reconfiguration procedure. The OBC may send control SVPs to terminals to notify the outcome of the switch reconfiguration.

(In real operation, the OBC may handle performance monitoring and traffic management functions. Storing of measurement data may be necessary such that the data can be sent back to the ground station for further processing. Traffic management functions include a) capacity allocation, where virtual path (VP) connections are semi-permanently allocated and virtual channel (VC) connections are on-demand allocation, b) call setup and release, and c) connectionless services such as switched multimegabit data service (SMDS) and connectionless broadband data service (CBDS).)

After the features have been demonstrated, the switch performance must be measured. Although there is no final recommendation on the switch performance, general guidelines were provided in Reference 4-1. The switch throughput should be above 90%. The FPS will be used to provide services for both delay-sensitive circuit traffic and loss-sensitive data traffic. The PLR and switching delay jitter should be kept small. The PLR should be between 10^{-9} and 10^{-10}. The switching delay (jitter) should be less than 0.4 ms. Interactive speech service demands the shortest end-to-end delay (30 ms). Clearly 30 ms end-to-end delay requirement is only for terrestrial network and is not applicable for satellite network. Data and compressed video traffic demands the lowest end-to-end packet loss ratio (10^{-10}). Another set of packet delay variation was recommended in Reference 4-2. The packet delay variation for voice and video is less than 6 ms and for data is less than 600 ms.

Section 5

Testbed Configuration

A block diagram of the fast packet switch testbed along with development support modules is shown in Figure 5-1. The key features of the testbed include switch and interface port operation at 155.52 Mbit/s, the use of commercial VLSI switching devices, two pairs of input and output processors for satellite virtual packet handling, congestion control based on priorities, and flexibility to accommodate additional features as needed in the future. As a part of the architecture definition task, a preliminary design of the testbed is presented in this section. The proposed design will be refined in the following high-level design task to finalize the overall testbed configuration.

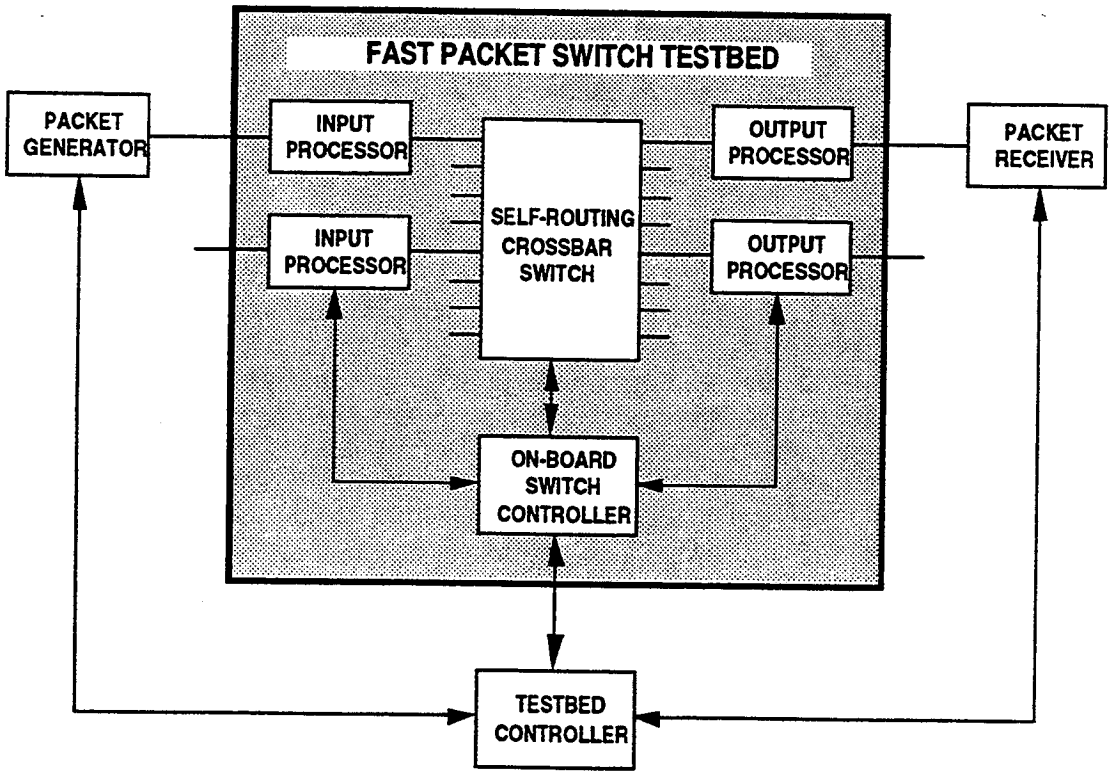


Figure 5-1: Fast Packet Switch Testbed Block Diagram

Figure 5-2 depicts a equipment configuration of the testbed and development support modules that will be used in the fabrication and testing of the switch.

The testbed will consist of a single chassis containing two input/output ports, a switch module, and a single board testbed control central processor unit (CPU). The boards located in the enclosure will have a 6U x 280 mm form factor. Depending upon the final results of the high level design, the switch input and output ports may consist of one or two

boards, the switch module will be a single board and the Control CPU will be a commercial SBE VLANE 68020 microprocessor. Thus a total of three to five wire wrap modules will be fabricated. The modules will contain ECL and TTL technologies with all ECL technology being restricted to the input and outputs of the port line cards, the switch interconnect and the switch real time control. The TTL technology used will make maximum use of high density programmable logic devices.

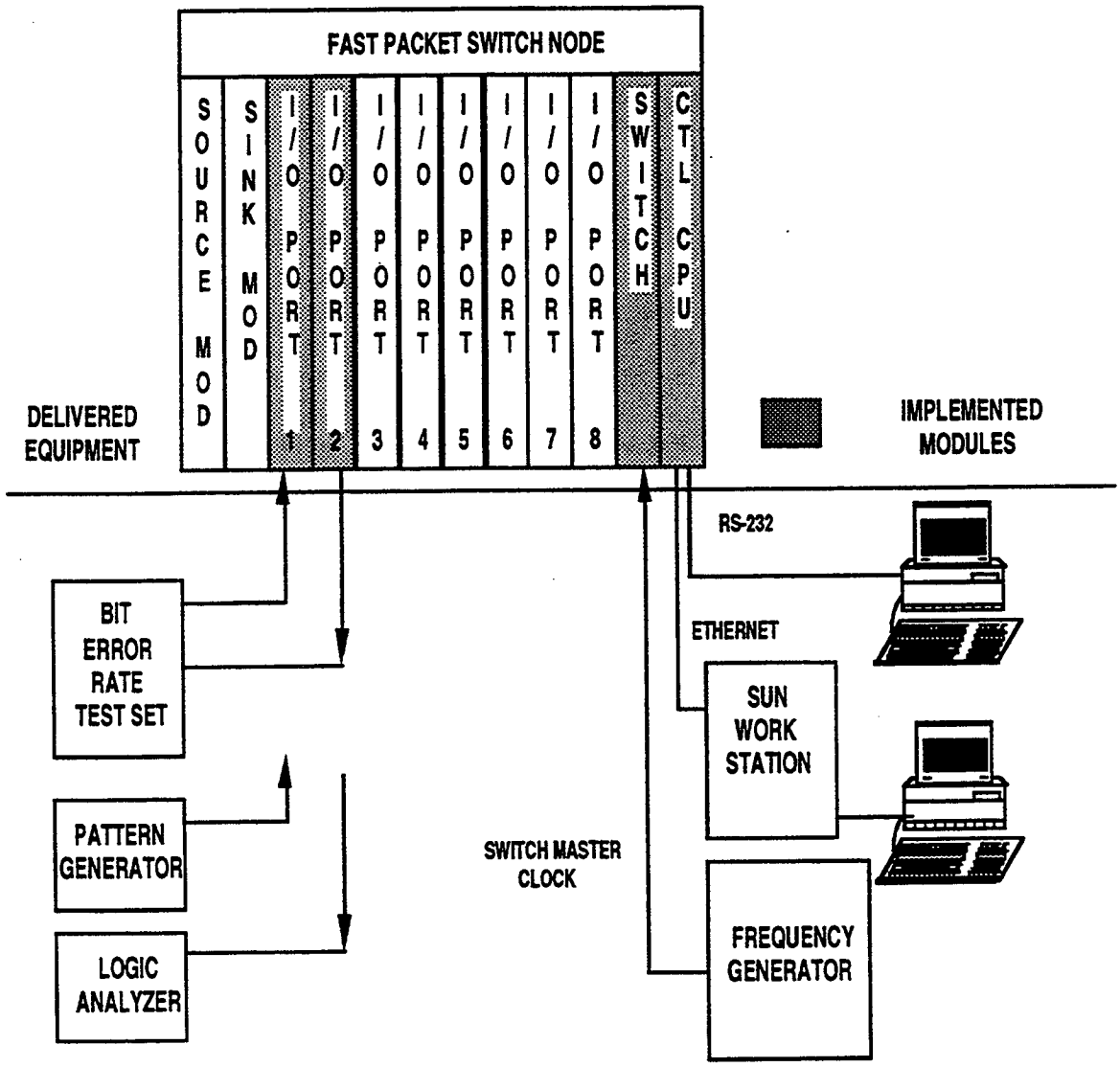


Figure 5-2: Testbed and Development Support Module

The testbed control processor will communicate with the various other modules via the P1 connector of the VME interface. All modules except the testbed CPU will act as bus slaves and their resources will be accessed as memory mapped IO. The following subsections describe the functional operation of the various modules and describe some of the capabilities being considered for fabrication to improve the switch testbed.

As shown in Figure 5.2 the development of a traffic generator source and a traffic generator sink module are also being considered as enhancements to the switch testbed. These modules would allow greater control over the possible traffic sources and sinks that may be used to verify the testbed operation and to measure its performance. This could also result in a better mechanism to test and measure different algorithms that may be used to improve the switch performance. Capabilities of such modules are described in the packet generator and sink subsections.

5.1 Packet Generator/Sink

Depending upon the final implementation the generation of traffic to test the switch operation and performance may be fairly straight forward or may require the development of some special assemblies. For the present system considered the tests are broken down into those functions that are necessary for the system operational verification and those that may be used to gain measures of the systems overall performance under various test scenarios. Mechanisms of generating and checking traffic flowing through the switch for these two classes of tests are described in the following paragraphs.

The operation of the test bed switch may be tested in a fairly simplistic manner by the use of pattern generators and logic analyzers. These tests will involve programming a canned pattern, corresponding to one or more satellite virtual packets (SVPs), into the pattern generator. These patterns may then be transmitted to one or more of the input ports. The logic analyzer may then be used to capture and scan the resulting data delivered from the output port or from any possible input port loopback connections. These tests are quite static and repetitive in nature and may be used in the system even if the data delivered at the output module undergoes some form of data conversion in passing through the switch (eg FEC encoding,CRC generation, different uplink and downlink rates and/or formats).

For those cases where the uplink traffic is of the same format as the down link traffic, certain test equipments (such as Packet BERT-200 from Microwave Logic) may be used to test the systems operation. It is also possible to program this equipment from a remote computer system via its GPIB or its RS432 interface to yield a more exhaustive set of tests. For those cases where the down link traffic is different either in format, rate or in various fields (possible by the insertion of error check sums or FEC code patterns) these test equipments may not be adequately suited for testing the switches operation.

For operational and systems performance testing more exhaustive exercising of the testbed may be achieved by developing some traffic generator and sink modules that could be used in conjunction with the control processor (and possibly an external computer like a sun workstation) for setting up various traffic generation and testing scenarios The basic functions required of such a traffic generator are to generate the the require uplink frame formats, to generate possible switch routing paths and to generate the data the will be contained in the SVPs. A block diagram of such a traffic generator and sink is shown in Figure 5.3.

The basic premise of the traffic generation/sink module is to allow a user to set up certain test configurations for the delivery of packet to the switch and the reception of packet from the switch. The packet is transmitted from and stored in (the transmit and receive) dual port RAMs located on the module. These RAMs could be implemented as ping-pong memories allowing one configuration to be tested while a new one is being loaded. The basic switch over of the active RAM to the standby RAM could be under the control of the testbed CPU, with the actual switch over occurring only on SVP boundaries. Synchronization between the traffic generator/sink module and the testbed control CPU could be via hardware interrupts and or via some form of polling mechanism. For packet delivery to switch there are three possible fields that must be accounted for.

The first field is the uplink synchronization field. Typically this field may contain some form of frame marker. Additional information may be required to accommodate specific uplink characteristics like the phase ambiguity resolution associated with QPSK modulation. This may be required for those cases where the ambiguity has not been removed by the demodulator. The uplink synchronization field is usually fairly small and remains unchanged for all frames. It may also be used for establishing appropriate SVP delineation boundaries. As mentioned in the previous sections an alternative mechanism could use the the SVP header and some form of forward error check sequence (as in ATM headers) for packet delineation. Here too the functions like demodulator phase ambiguity removal must be addressed. This may increase the overall synchronization time for the input module or may increase the amount of synchronization hardware required. For the present implementation it is expected that packet delineation will be performed by a specific frame sequence added to the front of each SVP.

The second field that must be accommodated for is the SVP header. This field will contain the various overheads associated with the SVP. In particular this field will contain the physical routing tag needed to route the SVP through the switch fabric. This field may also contain a FEC code necessary to protect the SVP header. These codes may be precalculated or additional hardware may be added to generator/sink modules to generate or check them on the fly. The final field that must be accommodated for is the actual SVP payload.

A possible implementation of the traffic generator/sync module would permit each of these three fields to be controlled separately, allowing the overall processing required by the testbed control processor to be minimized for different types of test and allowing the dual ported RAM to be used most effectively. A couple of data memory maps that could be used in such an implementation is also shown in Figure 5.3 The first map would allocate a fixed area (say 16 words or less) to the uplink synchronization field. This field will be transmitted every frame. The second area of RAM could be used to program the SVP payload. For the case shown this payload would be the same for each SVP. Alternatives could allow the payliad to be a PSR or a simple cyclic counter. The third area of the RAM could then be allocated to various routing tags and SVP headers. This would allow a large number of packets to be generated each going to a different possible output port. An alternative mapping also shown would allow the generation of a smaller number of SVPs each with a different SVP payload.

An additional capability for the traffic generator could allow any or all of the various fields to be generated from an external source like a bit error rate test set with the sole restriction being that the external source must be capable of operating under the gated control of the traffic generator. This gated control may be accomplished by controlling the clock to the external source or by supplying a clear to send signal to the device.

Packet reception of the sink module would involve the storage of packet into the sink dual port RAM or the sending of one or more of the above mentioned fields to an external device. (like the receive side of a bit error rate test set.)

The proposed module would interface to the testbed control processor via the VME interface bus. This control processor could have its various test patterns generated via a console connected to the RS-232 port or be downloaded from a workstation via the RS-232 or the the ethernet port. When the control processor receives the configuration data it then proceeds to write it into the transmit dual ported RAM located on the source module and to setup the particular test. It can then activate either side of the transmit or receive ping-pong memory. Further enhancements may include mechanisms of specifying how many times the ping pong buffers should be transmitted or received prior to informing the control processor the test is completed or to switching to the alternate memory.

5.2 Input/Output Port

The input/output port will be responsible for accepting incoming data, establishing proper uplink synchronization, demultiplexing the data, and preparing it for delivery to the switch module for routing to the appropriate output module. At the output module, the data is accepted and possibly reformatted for delivery to the appropriate down link.

A block diagram of the input/output port module is shown in Figure 5.4. Modifications of the design being considered from that of the initial proposal are to allow the various input and output ports to transfer data at rates of about 155.52 Mbit/s and to use a commercially available high speed conventional crossbar switch chip in the switching fabric. To accommodate these modifications and to minimize their effects the proposed design tries to restrict the number of areas requiring high speed technology to a minimum and further requires that all processing of various data fields be done on byte wide boundaries. Also shown in Figure 5.4 in the lightly shaded areas are two additional functions that may be required for on-board switching but that have not been included to date in the baseband switch testbed. These functions involve uplink and down link Reed Solomon encoding and down link data interleaving to randomize the effects of any burst of errors on the down link encoding. The addition of these functions could be included at a later time.

As shown in Figure 5.4, data is transferred to and from the port line card over two differential serial lines at rates of 155.52 Mbits/sec. It should be noted that higher data rates may be required if the uplink data were encoded prior to transmission. The dark portions of the figure indicate those areas of the design that must be implemented in emitter coupled logic (ECL). Upon reception of the ECL data the input portion of the

module quickly converts the incoming data into 8 bit bytes. This data is then delivered to the SVP Sync and Dmux Timing logic (SSDT) to establish frame/byte synchronization.

This frame/byte synchronization is tied to the overall acquisition algorithm implemented by the SSDT. The SSDT scans all bytes of data to try and identify the first byte of a multiple byte frame pattern. If this pattern is not detected within a certain number of frames the SSDT causes the uplink converter logic to execute a bit strobe clock inhibit into the serial to parallel converter. This causes a single bit rotation in the succeeding 8 bit bytes of data generated. The SSDT then proceeds to scan another set of frames for the appropriate byte pattern. Upon detection, the succeeding bytes of data are tested to see if they correspond to the multi-byte pattern expected. If they do not the inhibit signal is regenerated and the logic again searches for the first byte of the frame pattern. This search continues until the whole frame pattern is detected. With the pattern detected the SSDT then proceeds to reload its frame counter and scan "N" of the next succeeding frames to insure that the frame marker is in the expected location. If the marker is not there for any one of "N" frames the SSDT reverts to the execution of inhibiting the clock to the serial to parallel converter. If the frame markers are detected for "N" frames in a row then the SSDT declares that line synchronization has been achieved. With synchronization achieved the SSDT continues to scan succeeding frames for the frame marker. Synchronization is declared loss and the above cycle repeats itself if "M" frames are detected in a row with an incorrect frame marker. The counter values of the initial frame byte search, the acquisition frame count "N" and the synchronization frame loss "M" may be programmable up to a certain limit (possibly 16).

With synchronization achieved the SSDT may then proceeds to demultiplex the incoming SVPs and deliver them to the switch input port logic (SIPL). The SIPL accepts byte data from the SSDT, processes the necessary header information and stores the SVP in the appropriate memory locations of a dual ported memory that is used as an input queue. The header processing being considered for this implementation involves service type (circuit or packet SVP) and packet priority (two possible priorities) for single-size SVPs. SVPs with different service types and/or priorities will be routed to different queues each with their own specific queue scan depth. The implementing effects of multi-size SVPs and of header FEC protection/correction will also be reviewed. The queue implemented will be capable of storing up to 128 SVPs. Attempts will be made to modularize the queue management hardware such that alternate memory managements mechanisms (for example preallocated areas per service type/packet priority or some form of link list implementation) may be tried at a future date. The present implementation being considered also assumes that each SVP delivered to the SIPL contains a physical routing tag that is used to route the packet through the switch fabric. This tag identifies which output port(s) the packet is to be delivered to.

In addition to storing the data in memory the SIPL must perform two additional functions. These are switch output port contention resolution and delivery of packet to the switch module itself. The first function is implemented to allow the various output modules to reserve a connection to one or more output ports. The basic operation of the contention control follows a similar methodology as identified in the SCAR I final report with a number of contention frames being transferred between the various input modules over the contention ring each SVP timeslot. This number will correspond to the total input queue scan depth selected (scan depth of all queues).

Since the present system design will accommodate different service types and priorities and will use of a commercial crossbar switch IC for module interconnection, the format of the contention control messages flowing through the input port will be modified. The crossbar switch ICs uses a centralized control port to establish switch connectivity and is not self routing on each of its input ports. As such the contention control format must be modified so that at the end of the contention cycle the centralized controller may use the results of the contention algorithm to setup the switch connectivity for the next SVPs. This requires the input ports specify their address in the appropriate desired output port reservation field(s). In addition for the input ports to properly reserve a connection for a SVP of a particular type/priority the format may require individual input ports to specify the SVP priority and service type. This information will be used by the various input modules to establish switch connectivity. This is in contrast to the format used in the SCAR I report where a single bit could be used for each switch connection point. As the switch control port is doubly buffered the contention control algorithm will be establishing switch connectivity for one SVP in advance.

With the input port contention resolved for the present SCP time slot, the SIPL then proceeds to transfer packet through the switch fabric to the reserved output port . Packet delivery to the switch module is fairly straight forward. This transmission is synchronized by the switch module itself via the "NEWSVP" and "SVPHEADER" interface signals. Upon activation of these signals the SIPL proceeds to transmit its data to the parallel to serial ECL conversion logic used to feed the switch fabric. The 8 bit data delivered and is then transmitted over a serial differential line to the switch module.

Packet reception from the switch module is also under the synchronization of the switch module and is also via a serial differential ECL drivers. Once again the NEWSVC and SVCHEADER signals are used to properly window the data as it is delivered to the output port. These signals are also used to establish proper byte alignment at each of the various output modules. It is expected that the switch output port logic will be very similar if not identical to the switch SIPL with the exception that the contention control port will not be utilized. Packet received by the switch output port may be stored in a dual ported RAM prior to delivery to the Mux & Timing Control (MTCL) logic. This data storage is only necessary if the switch fabric operates faster than the output port line rate or if the output port may be expected to service a number of down link carriers or operate in a spot beam environment. For these cases the SVP header will again be processed to route the incoming data to the correct queue. Hardware counts may also be kept as to number of cells in a queue or to the cells loss due to lack of storage space for data of a specific service type or priority. These counts may be read by the control processor.

The MTCL is responsible for reformatting the output data prior to transmission. If the uplink synchronization code is not striped off at the switch input module and is the same as that used in the down link this block of logic may quite simple in that it need only control the parallel to serial conversion to the output port line side drivers.

5.3 Switch Module

A block diagram of the switch module is shown in Figure 5.5. The switch module utilizes a commercial crossbar switch IC with additional control logic that can allow the various input modules to utilize the switch in a self routing fashion. The crossbar switch itself will be a single chip device implemented GaAs or Bipolar technology. All interfaces to the chip will be ECL compatible. The crossbar switch will be capable of accepting serial data at rates of 155.52 Mbit/s. The overall control of the switch connectivity will be via TTL logic which interfaces the contention control logic to the switch control logic. As such the switch module acts as the starting element on the token ring. The logic acts as a simple repeater as the various modules contend for their desired output connectivity and at the end of the contention cycle the switch logic extracts the resulting connectivity information and uses it to configure the crossbar switch. At the appropriate time the control logic activates the specified connectivity map. This map is activated such that the connectivity for all input modules switch at one time. An additional function of the switch control logic is to generate the appropriate synchronization signals for the switch input and output ports so that the data to and from the switch is properly bit and byte aligned.

As shown in the diagram the switch control logic would also have an interface to the testbed control processor. This interface may be used to configure any local variables or as a possible bypass mechanism to the contention control logic to allow the switch connectivity to be programmed in a static manner. Such capabilities may be useful debug aids.

5.4 Control Processor

The control processor used in the switch test bed will be commercial 68020 SBE VLANE microprocessor. The Processor will have RS-232 and ethernet connection capabilities. The processor will have a system console process which will allow the operator to perform various basic functions over the VME bus. These functions include reading and writing the various memory devices and memory mapped IO control registers. For the case where traffic generator and sink modules are developed the control processor can also be used to configure the traffic generator and monitor the traffic sink operations. For certain cases this data may be loaded over the ethernet port and may be activated via the ethernet port.

5.5 Implementation Approach

The present implementation approach will use the same methodology as outlined in the SCAR II proposal with the exception that the port interface data rates will be much higher (150 M to 200 M from 20 to 30M) and that the switch implementation will use commercially available crossbar switches whose output connectivity may be controlled by a centralized contention control algorithm that is programmed by the various input modules themselves.

To accommodate these modifications and to minimize their effects the proposed design tries to restrict the number of areas requiring high speed technology to a minimum and further requires that all processing of various data fields be done on byte wide boundaries. It is estimated that the ECL interfaces on the input module will require the use of about 40 small scale and medium scale ECL integrated circuits. In addition the switch module itself will require about 20 extra ICs to act as ECL drivers/receivers and ECL/TTL converters. The overall designs required however are relatively straight forward though extra care will have to be made in the board layout and fabrication.

The remaining logic of the switch testbed will make maximum use of high density TTL compatible programmable logic like the ALTERA EPM 5000 and 7000 series chips which have system clock rates of about 40 to 50 MHz. Interfaces to and from these chips will accommodate byte or multi-byte wide data processing. A standard commercial enclosure will be used to house the testbed and commercially available boards will be use for module interconnect.

Where possible the various parameters used to define the SVP format, the acquisition and synchronization algorithm and the traffic source sink capabilities will be made programmable via the testbed control processor. By these mechanisms the testbed can be made more flexible and can be used to study a wider variety of operational characteristics. In addition by the use of modular design techniques attempts will be made to allow the hardware and and software developed for the testbed to be readily expanded and or enhanced to allow for functions not presently included in the design or possibly not yet specified. These capabilities may include additional hardware software hooks for the testing of alternate contention control or congestion control algorithms, queue management algorithms or for the incorporation of different type of redundancy development.

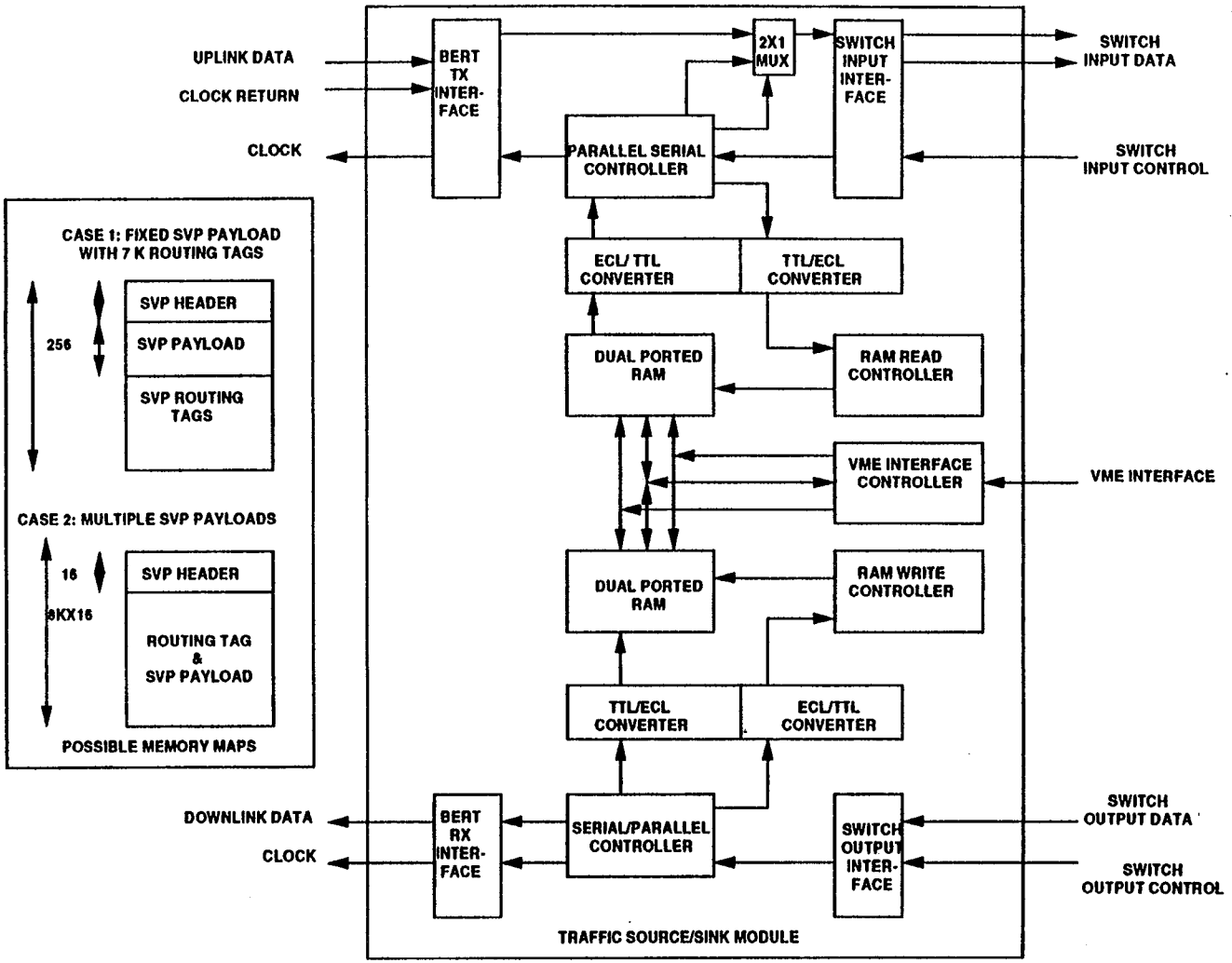


Figure 5-3: Traffic Source/Sink Block Diagram

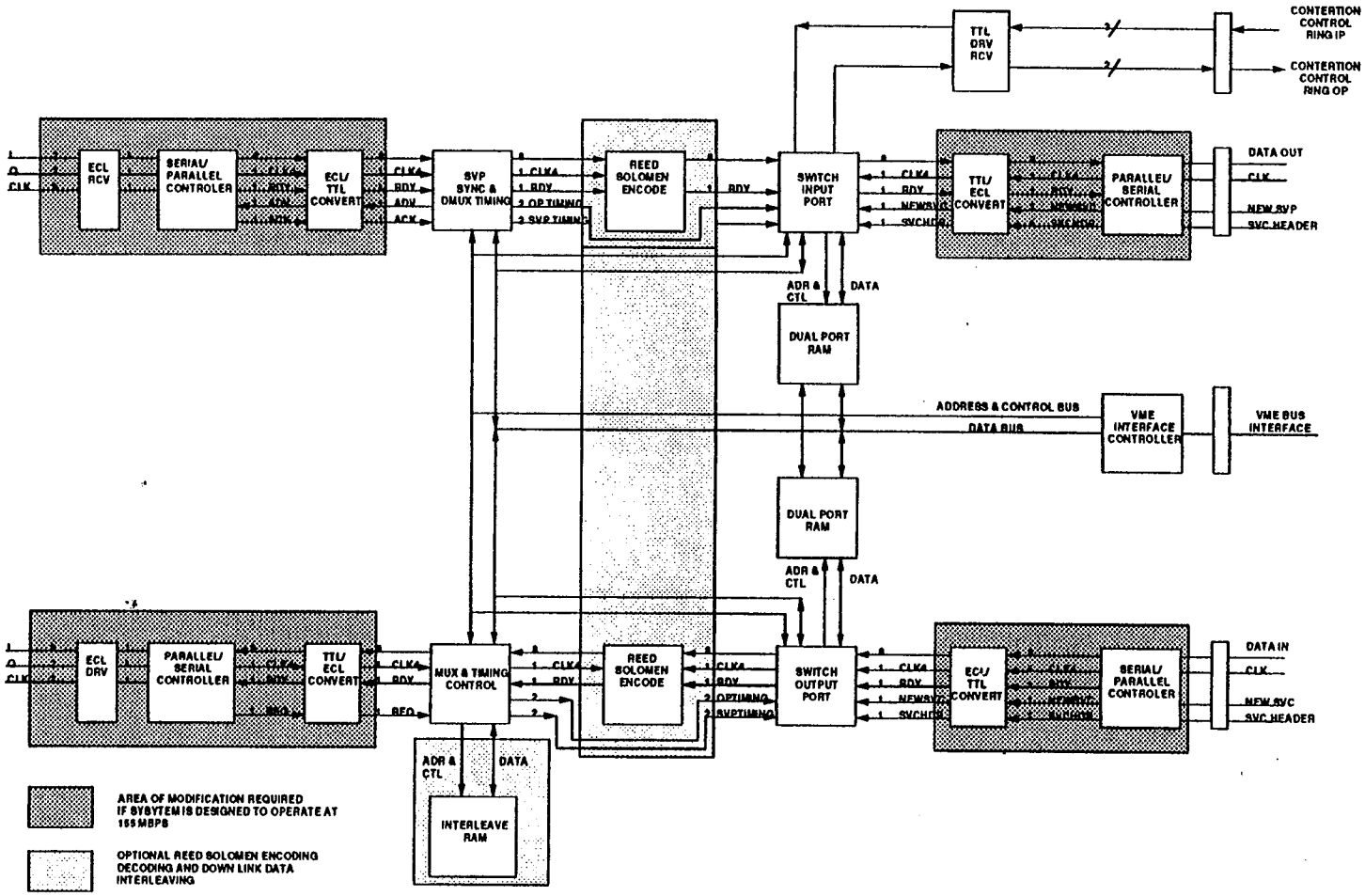


Figure 5-4: Testbed Switch Input/Output Port

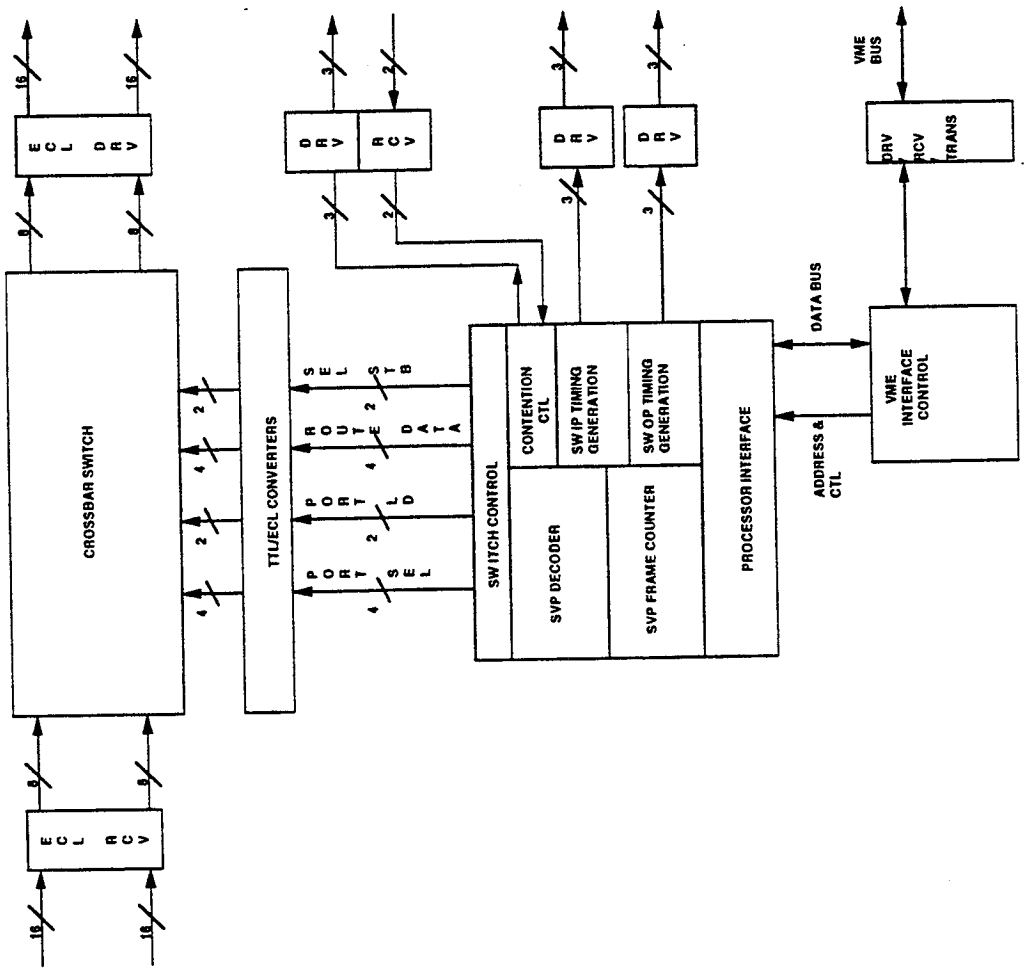


Figure 5-5: Switch Module

Section 6

Preliminary Test Plans

This section presents the system-level test plans for the on-board FPS. These test plans are only for the breadboard under development. In real system, more extensive testing is required. The basic functions of the FPS should be tested, verified, and demonstrated. After the switch operations have been tested to be satisfactory, the next step is to measure and analyze the switch performance. In general testing the FPS should follow the sequence: 1) module acquisition, 2) synchronization, 3) packet queueing, 4) token ring generation, 5) output port reservation, 6) packet transfer, 7) packet reception, 8) packet storage, 9) packet formatting, and 10) packet end to end performance measurements. Some of the preliminary test procedures are discussed at the end of this section. Detailed test procedures will be presented in the "Test Plan" task.

6.1 Input Port

The basic functions of the input port, which should be tested, contain:

- synchronization

The input port should demonstrate the capability of identifying packets' boundary from the demodulated bit stream.

- packet queueing

The arrival packet must be inserted to the right location of the right queue based on the payload type and the loss priority.

- output contention resolution

The output contention resolution module must resolve output contention among different packets at different input ports. The input port should demonstrate the capability of reserving a subset of the destinations of a multicast packet, i.e., the input port should have the call splitting capability. The fairness of output contention resolution among different input ports must be achieved. Loss of tokens must be detected.

- priority control for integrated operation and guaranteed QOSs

Priority control is used in two areas in a fast packet switch: multiplexing sequence and congestion control. Priority is used to meet different packet loss ratio requirements and switch delay requirements. Circuit switched traffic is delay sensitive and packet switched traffic is delay insensitive. The circuit switched traffic has a higher priority than the packet switched traffic. Within the packet switched traffic, there are two more subclasses: loss sensitive and regular. A high-priority packet is guaranteed to win the output contention

resolution when it is contended with other low-priority packets. Also the low-priority packets are dropped before the high-priority packets when the switch is congested.

• header error control (option)

To ensure the arrival packets are error-free, the HEC function should be performed at the input port. By performing HEC, the bandwidth efficiency is increased since the erroneous packets are discarded at the very earliest stage.

• traffic monitoring

Traffic monitoring is a necessary procedure to detect congestion. The input port should provide the switch loading (such as arrival rate, queue length, or output utilization) to the congestion control algorithm. More discussion on congestion control is provided in the "Critical Design and Simulation" task.

• idle SVP discarding

Idle SVP should be discarded. Saving of the bandwidth can be used to transmit the control SVPs originated from OBC.

6.2 Switching Fabric

The basic functions of the switching fabric, which should be tested, contain:

• self-routing and packet transfer

The FPS should route packets to the destination solely based on packets' routing tags.

• connection-oriented switching

The FPS can not transmit packets with the same virtual channel number (VCN) out of sequence. In other words, the transmission sequence of the arrival packets have to be preserved.

• multicast/broadcast

A multicast FPS should route a multicast packet from one input port to multiple output ports. The FPS must demonstrate the capability of duplicating a multicast packet to multiple copies and send these copies to the destinations.

6.3 Output Port

The basic functions of the output port, which should be tested, contain:

• traffic monitoring

Traffic monitoring is a necessary procedure to detect congestion. The input port should provide the switch loading (such as output utilization) to the congestion control algorithm. More discussion on congestion control is provided in the task "Critical Design and Simulation".

- speed/format conversion

Since the downlink and the uplink may use different speed and/or transmission format, the speed/format conversion function has to be tested.

- routing tag removal (option)

Since the routing tag is overhead, the routing tag may be removed from the packet header at the output port. The saving of bandwidth can be used to transmit the control packets originated from the OBC.

- idle SVP insertion

To maintain the downlink synchronization, idle SVPs are inserted into the downlink TDM stream when there is no traffic to be sent.

6.4 OBC

The basic functions of the OBC, which should be tested, contain:

- fault tolerant operation

The fault tolerant operation consists of fault detection, fault diagnosis, fault isolation, and fault reconfiguration. To test the fault tolerant operation, for example, an output port can be put in a faulty state. If the fault tolerant operation is successful, the switch should reroute the incoming packets (destined to the faulty output port) to a redundant output port.

- congestion control

The OBC receives the switch loading information from the traffic monitoring devices (at the input ports and/or the output ports). The OBC performs congestion control algorithm. The OBC sends out congestion control messages to the ground terminals at regular intervals.

After all the characteristics and features mentioned above have been tested, the performance of the switch should be measured and analyzed. The switch itself can be characterized by the throughput and switch capacity. The connection can be characterized by bit error rate (BER), instaneous and long term packet loss ratio, packet (mis)insertion ratio, packet transfer delay, and packet delay jitter. Integration of circuit switched traffic and packet switched traffic can be tested by measuring the packet delay jitter of the circuit switched packet. The effectiveness of the congestion control algorithm can be tested by measuring the on-board switch PLR when the switch is overloaded.

If budget permits, it may be desirable to compare the measurement results (such as PLR) with analytical results or software simulation results. The comparison results can also be used to verify the switch operation.

6.5 Preliminary Test Procedure Considerations

In the SCAR II design proposal, methodologies of generating test conditions mainly consider the extensive use made of pattern generators and logic analyzers to act as packet sources and sinks. Bit error rate test set could also be used to gain a measure of the overall hardware performance. Additional test capabilities that are presently being reviewed include various loopback capabilities in the input/output port modules. These loopbacks may include near end loopbacks before and after the ECL line interface, loopbacks before the switch module and far end loopbacks through the switch module itself. Additional capabilities may be built into the switch module are to disable the real time switch controller and to allow the CPU to have access to the switch control for static establishment of switch connectivity paths. Additional test capabilities may include the ability for scan registers within the various ALTERA designs and the ability to have the control CPU readback the various registers used to configure the testbed. These read and write capabilities could be executed from the testbed control processor via the RS232 port. Additionally the ethernet interface may be used to down load various configuration and test scenarios from a remote host computer. Extensive use of this interface will be made in the downloading of code to the testbed control CPU during its initial debug.

Some of these tests will depend upon the design of the packet generator and sink modules and the features incorporated within this module. The total capabilities are presently being reviewed in light of the present budget and manpower restraints imposed by SCAR II proposal and the additional enhancements necessary allow the testbed to operate at rates of about 155.52 Mbit/s.

Section 7

Conclusions

The design principles of different multicast switching architectures are reviewed, which include a summary from Phase 1 report and some new multicast switching architectures. Commercially available switching chips are surveyed for potential space applications. The multicast crossbar switch is selected for subsequent breadboard development. The multicast crossbar switch has the following advantages: it is commercially available, its structure is simple, the switching fabric is point-to-multipoint nonblocking, and the operation characteristics (such as power) are very suitable for on-board applications. Different queueing approaches are reviewed. The input queueing strategy is selected for easy implementation and low complexity. The selection matches with the recommendation made in Phase 1 report.

Due to head of line blocking at the input queue, the packet switch throughput (for point-to-point connections) can not exceed 58% for a larger N. To increase the switch throughput, two efficient scheduling algorithms are identified. The first algorithm is to use the basic centralized ring reservation scheme with a large checking depth. The second algorithm is to use the centralized ring reservation scheme with future scheduling. The final selection will be determined at the "High Level Design" task.

The header of SVPs contains the routing tag and other satellite network internal fields such as payload type and QOS. The routing tag is used to route through the on-board switch. The routing tag is inserted in the SVP header at the earth station. For ATM application, the VPI has a local significance in the satellite network. The VPI needs to be retranslated at the earth station. However, no VPI retranslation is required at the on-boar switch. Grouping of cells (or other types of traffic) should be based on the downlink beam if there are a large number of terminals in the network. Grouping of cells (or other type of traffic) should be based on the receiving earth station if there are a few, large earth stations in the network. If single-size SVP is chosen for Phase 2 development, the SVP size should be less than or equal to that of 4 cells. If the traffic foreseen is very diverse, then multiple-size SVPs should be considered. There are four different sizes: single-cell SVP, 2-cell SVP, 4-cell SVP and 8-cell SVP. Two synchronization schemes are proposed. The first follows synchronization method used in the TDM frame synchronization and a frame format is required. The second follows the techniques used in the ATM cell header error control synchronization (ATM cell self-delineation) and no external frame format is required. For multicast multiple-size SVPs, the switch operation should allow call splitting and enforce continuous transmission for the SVP packet through the switch. Final selection of SVP format and synchronization scheme is determined at the "High Level Design" task.

Two types of priorities are proposed for subsequent development. The first is based on the service type and the second is based on the loss priority. The service type priority is used to distinguish the arrival packets are circuit switched or packet switched traffic. The loss priority is applied to packet switched traffic only. The total number of priorities is three: circuit data, high-priority packet data and low-priority packet data. There are three logical subqueues at each input port, where one for each priority. When the packets arrive to the

input port, the input port examines the QOS field and insert the packet to the proper subqueue. The insertion/removal of the packets to/from the subqueue should be implemented in a link list fashion. There is no upper limit for the circuit switched data queue length and the high-priority queue length, but a limit is set for the low-priority queue length. The basic centralized ring reservation scheme is modified to accommodate packets with different priorities. The token streams will be sent to the input port three times for three priorities. At each time, only the packets with the corresponding priority can reserve the tokens (output ports). The checking depth for each subqueue will be determined based on the traffic amount for each priority; the summation of the checking depth of the subqueues at an input port is a constant.

Integration of circuit and packet switched traffic adopts the switch capacity reservation scheme. The system capacity for circuit connections in one earth station is reserved each frame. The circuit packets can be transmitted at any assigned slots at the sending terminal in one frame. However, the circuit SVPs need to participate in output contention. The delay jitter of circuit connections is bounded using priority control. A small amount of queueing delay is experienced at the switch. A smoothing buffer is required at the receiving terminal to compensate the delay jitter.

Different fault tolerant designs are investigated for the FPS. For the purpose of demonstration, 1-for-N redundancy for input port, switching path, and output port will be incorporated into the breadboard design. Software is built into the OBC to perform fault detection, fault diagnosis, and fault reconfiguration.

Switching subsystem high-level functional requirements are identified. These requirements serve as a functional specification for the "High Level Design" task. A preliminary testbed configuration and test plans are also provided. The basic functions of the FPS should be tested and verified. And then the switch performance should be measured and analyzed.

Section 8

References

[1-1] L. Adams, "The Virtual Path Identifier and its Applications for Routing and Priority of Connectionless and Connection-Orientated Services," International Journal of Digital & Analog Cabled Systems, vol. 1, no. 4, pp. 257-262, 1988.

[1-2] M. Prycker, "ATM Technology: a Backbone for High Speed Computer Networking," Computer Networks and ISDN Systems, vol. 25, pp. 357-362, 1992.

[2-1] On-Board B-ISDN Fast Packet Switching Architectures - Phase 1: Study -, Final Report, NASA Contract NASW-4528 , Prepared by COMSAT Laboratories, Jan. 1992.

[2-2] J. Hayed, R. Breault, and M. Mehmet-Ali, "Performance Analysis of a Multicast Switch," IEEE Transactions on Communications, vol 39, no. 4, pp. 581-587, April 1991.

[2-3] X. Chen and J. Hayes, "Call Scheduling in Multicasting Packet Switching," ICC, pp. 895-899, 1992.

[2-4] J. S. Turner, "Design of a broadcast packet switching network," IEEE Transactions on Communications, vol. 36, no. 6, pp. 734-743, June 1988.

[2-5] T. T. Lee, "Nonblocking Copy Network for Multicast Packet Switching," IEEE JSAC, vol. 6, no. 9, pp. 1455-1467, Dec. 1988.

[2-6] D.-J. Shyy, "Nonblocking Multicast Fast Packet/Circuit Switching Networks," COMSAT Invention Disclosure No. 31-E-10, June 1991.

[2-7] Y. Yeh, M. Hluchyj and A. Acampora , "The Knockout Switch: A Simple Modular Architecture for High-Performance Packet Switching," IEEE JSAC, vol. 5, pp. 1274-1283, Oct. 1987.

[2-8] R. Cusani and F. Sestini, "A Recursive Multistage Structure for Multicast ATM Switching," INFOCOM, pp. 1289-1295, 1991.

[2-9] K. Eng, M. Hluchyj and Y. Yeh, "Multicast and Broadcast Services in a Knockout Packet Switch," INFOCOM, pp. 29-34, 1988.

[2-10] M. Karol, M. Hluchyj, and S. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," IEEE Trans. on Communications, pp. 1347-1356, Dec. 1987.

[2-11] M. Hluchyj and M. Karol, "Queueing in High-Performance Packet Switching," IEEE Trans. on Communications, pp. 1587-1597, Dec. 1988.

[2-12] L. Kleinrock, "Queueing Systems Vol I: Theory," New York: Wiley, 1974.

[2-13] Congestion Control - Modeling and Simulation of Congestion Control in a Destination Directed Packet Switch for Satellite Communications -, Final Report, Prepared by COMSAT Laboratories, NASA Contract NAS3-25933, Nov. 1992.

[2-14] I. Iliadis and W. Denzel, "Performance of Packet Switches with Input and Output Queueing," ICC, pp. 747-753, 1990.

[2-15] Y. Oie, M. Murata, K. Kubota, and H. Miyahara, "Effect of Speedup in Nonblocking Packet Switch," ICC, pp. 410-414, 1989.

[2-16] J. Chen and T. Stem, "Throughput Analysis, Optimal Buffer Allocation, and Traffic Imbalance Study of a Generic Nonblocking Packet Switch," IEEE JSAC, pp. 439-449, April 1991.

[2-17] A. Pattavina, "A Broadband Packet Switch with Input and Output Queueing," ISS, 1990.

[2-18] K. Hajikano, T. Nomura, and K. Murakami, "ATM switching technologies," FUJITSU Science and Technology Journal, vol. 28, no. 2, pp. 132-140, June 1992.

[2-19] TQS Digital Communications and Signal Processing, TQ8016.B, March 1992.

[2-20] AMCC, S2024 Preliminary Device Specification, 1991.

[2-21] VSC, VSC864 Preliminary Data Sheet, Oct. 1991.

[2-22] P. Barri and J. Goubert, "Implementation of a 16 x 16 switching element for ATM exchanges," IEEE JSAC, vol. 9, no. 3, pp. 751-757, June 1991.

[2-23] T. Kozaki, and etal, "32 x 32 shared buffer type ATM switch VLSI's for B-ISDN's," IEEE JSAC, vol. 9, no. 8, pp. 1239-1247, Oct. 1991.

[2-24] "Science and Technology Highlights 1992," Toshiba Review, page 2, Aug. 1992.

[2-25] T. Itoh, and etal, "Practical implementation and packaging technologies for a large-scale ATM switching systems," IEEE JSAC, vol. 9, no. 8, pp. 1280-1288, Oct. 1991.

[2-26] T. Itoh, and etal, "Sunshine: a high-performance self-routing broadband packet switch architecture," IEEE JSAC, vol. 9, no. 8, pp. 1289-1298, Oct. 1991.

[2-27] H. Matsunaga and H. Uematsu, "A 1.5 Gb/s 8 x 8 cross-connect switch using a time reservation algorithm," IEEE JSAC, vol. 9, no. 8, pp. 1309-1317, Oct. 1991.

[2-28] W. Fisher and etal, "A scalable ATM switching system architecture," IEEE JSAC, vol. 9, no. 8, pp. 1299-1307, Oct. 1991.

[2-29] M. Schroeder and etal, "Autonet: A high-speed, self-configuring local area network using point-to-point links," IEEE JSAC, vol. 9, no. 8, pp. 1318-1335, Oct. 1991.

[3-1] Information Switching Processor Contention Analysis and Control Study, Final Report, NASA Contract NAS3-25933 , Prepared by COMSAT Laboratories, Jan. 1992.

[3-2] B. Bingham and H. Bussey, "Reservation-Based Contention Resolution Mechanism for Batcher-Banyan Packet Switches," Electronic Letters, vol. 24, no. 13, pp. 772-773, June 1988.

[3-3] H. Obara, "Optimum Architecture for Input Queueing ATM Switches," Electronics Letters, 28 th, pp. 555-557, March 1991.

[3-4] H. Matsunage and H. Uematsu, "A 1.5 Gb/s 8 X 8 cross-connect switch using a time reservation algorithm," IEEE JSAC, vol. 9. no. 8, pp. 1309-1317, Oct. 1991.

[3-5] M. Karol, K. Eng, and H. Obara, "Improving the Performance of Input-Queued ATM Packet Switches," INFOCOM, pp.110-115, 1992.

[3-6] U. Yeh, M. Hluchyj, and A. Acampora, "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching," IEEE JSAC, vol. 5, pp. 1274-1283, Oct. 1987.

[3-7] K. Toyoshima, M. Sasagawa, and I. Tokizawa, "Flexible Surveillance Capabilities for ATM-based Transmission Systems," ICC, pp. 699-704, 1989.

[3-8] J. Bae, and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks," Proceedings of the IEEE, vol. 79, no. 2, pp. 170-189, Feb. 1991.

[3-9] T. Lee, M. Goodman, and E. Arthurs, "A Broadband Optical Multicast Switch,", ISS, vol. 3, pp. 7-13, 1990.

[3-10] D.-J. Shyy, "Advanced On-Board Multicast Switching Architectures: Part 3 - Priority Control, " COMSAT Laboratories Technical Note, NTTR-165, Jan. 1992.

[3-11] W. Fischer and etal., "A Scalable ATM Switching System Architecture," IEEE JSAC, vol.9 no. 8, pp. 1299-1307, Oct. 1991.

[3-12] D. Rennels, "Distributed Fault-Tolerant Computer Systems," Computer, pp. 55-65, March 1980.

[3-13] R. Williams, B. Johnson, and T. Roberts, "An Operating System for a Fault-Tolerant Multiprocessor Controller," IEEE Micro, pp. 18-29, Aug. 1988.

{3-14] T. Ono and M. Mori, "On-board Satellite Switch Controller for Multi-beam Communication Satellite," ICC, pp. 1086-1090, 1990.

[3-15] J. Hayes and E. McCluskey, "Testability Considerations in Microprocessor-Based Design," Computer, pp. 17-26, March 1980.

[3-16] D. Pradhan and J. Stiffler, "Error-Correcting Codes and Self-Checking Circuits," Computer, pp. 27-37, March 1980.

[4-1] K. Hajikano, T. Nomura, and K. Murakami, "ATM Switching Technologies," FUJITSU Science and Technology Journal, vol. 28, no. 2, pp. 132-140, June 1992.

[4-2] K. Okada and etal., "A Study on Satellite-Switched TDMA Systems for Applying to the Asynchronous Transfer Mode," ICC, pp. 355-359, 1992.

Appendix A

Queueing Equation Derivation for Nonblocking Switch with Input Queueing

A.1 Nonblocking Switch with Input Queueing

This subsection derives the throughput of a nonblocking switch with input buffering. Assume the output contention resolution module chooses one packet among k packets destined to the same output randomly, where $0 \leq k \leq N$. The virtual queue concept is introduced first. If a packet is selected by the output contention resolution module, it will be transmitted to the destination at the next slot and will not enter the virtual queue. The virtual queue is used to hold the packets which loses the output contention resolution. An illustration of the virtual queue concept is shown in Figure A-1.

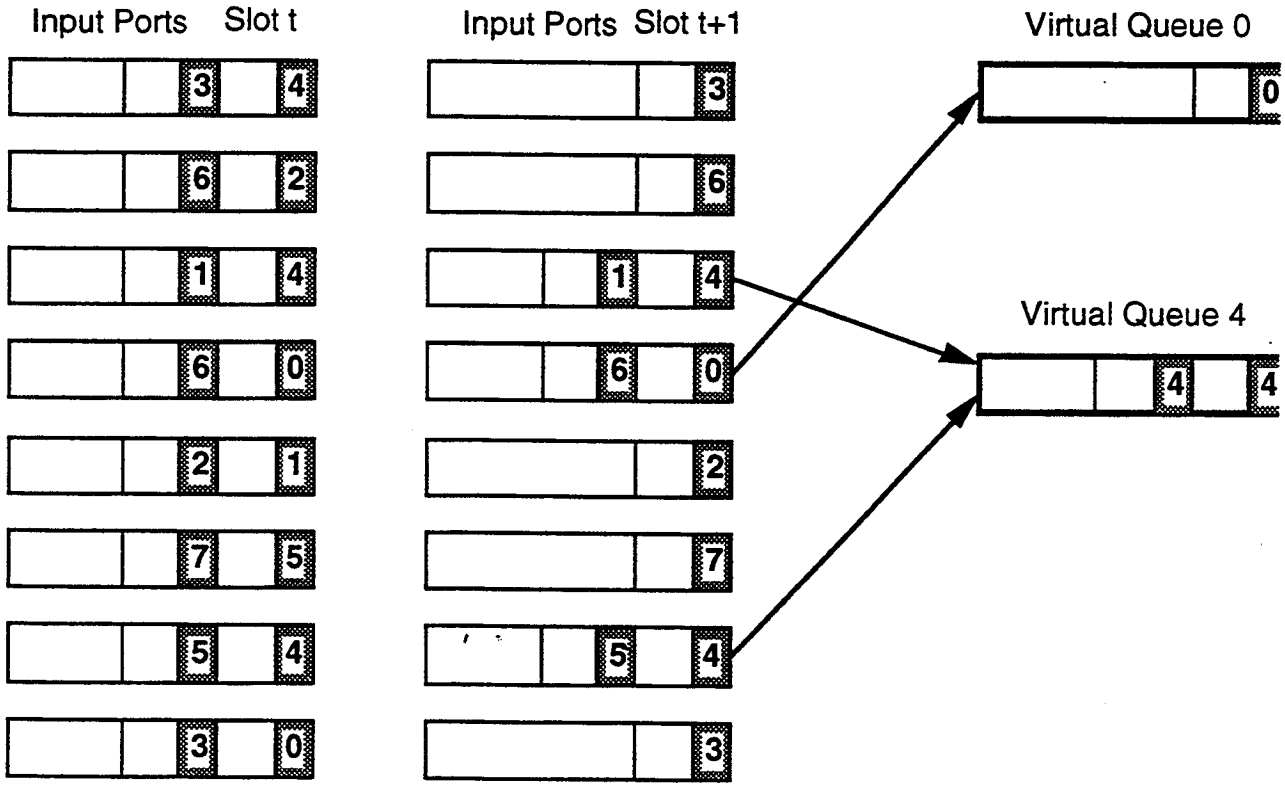


Figure A-1: An Illustration of Virtual Queue Concept

Define virtual queue i to consist of the HOL packets, which are not selected by the output contention resolution module, from different input queues destined to output port i. We analyze the behavior of the virtual queue i. Define V^i_m to be the number of packets in virtual queue i at the end of the mth slot. Define B^i_m to be the number of arrival packets to virtual queue i from different input ports during the mth slot. Define Fm to be the total number of packets left virtual queue i during the mth slot. By definition,

$Fm\text{-}1 = N - \sum_{i=1}^{N} V^i_{m-1} = \sum_{i=1}^{N} B^i_m$. Take expectation, $E[F] = N - N\,E[V]$.

Evidently, $\frac{E[Fm]}{N} = T$, where T is the throughput of the switch.

Therefore, $E[V] = 1 - T$. Assume the packet arrivals from different input ports to the virtual queue follows the Bernoulli process. Then

$$Pr[B^i_m = k] = \binom{Fm\text{-}1}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{F(m-1)-k}, \quad k = 0, 1, \ldots, Fm\text{-}1.$$

$$V^i_m = \max(0, V^i_{m-1} + B^i_m - 1)$$

$$\text{Define } \Delta_k = \begin{cases} 0 \text{ when } k = V^i_{m-1} + B^i_m = 0 \\ 1 \text{ when } k = V^i_{m-1} + B^i_m \geq 1 \end{cases}$$

Then $V^i_m = V^i_{m-1} + B^i_m - \Delta_k$

Use both sides of this equation as an exponent for z.

$$z^{V^i_m} = z^{(V^i_{m-1} + B^i_m - \Delta k)}$$

Take expectation: $E[z^{V^i_m}] = E[z^{(V^i_{m-1} + B^i_m - \Delta k)}]$

$$V_m^i(z) = \sum_{k=0}^{\infty} Pr\,[V_m^i = k]\, z^k = E\,[z^{V_m^i}] = E\,[z^{(k-\Delta k)}]$$

$$= \sum_{k=0}^{\infty} Pr\,[V_{m-1}^i + B_m^i = k]\, z^{k-\Delta k}$$

$$= Pr\,[V_{m-1}^i + B_m^i = 0]\, z^{0-0} + \sum_{k=1}^{\infty} Pr\,[V_{m-1}^i + B_m^i = k]\, z^{k-1}$$

$$= Pr\,[V_{m-1}^i + B_m^i = 0] + \frac{1}{z} \sum_{k=0}^{\infty} Pr\,[V_{m-1}^i + B_m^i = k]\, z^k - \frac{1}{z}\, Pr\,[V_{m-1}^i + B_m^i = 0]\, z^0$$

Note that $Pr\,[V_{m-1}^i + B_m^i = 0] = 1 - T$. Therefore,

$$V_m^i(z) = 1-T + \frac{1}{z}\,[V_{m-1}^i(z)\, B_m^i(z)] - \frac{1}{z}\,(1-T)$$

Let $m \to \infty$, $V^i(z) = \dfrac{(1-T)(1-z)}{B(z) - z}$

The average tagged output queue length can be derived using the property of z-transform.

$$E\,[V^i] = \frac{dV^i(z)}{dz}\,|_{z=1}$$

$$= \frac{(1-T)B^{(2)}(z) + (T-1)(1-z)B^{(3)}(z)}{2[B^{(1)}(z)-1]^2 + 2(B(z)-z)B^{(2)}(z)}\,|_{z=1} = \frac{N-1}{N}\,\frac{T^2}{2\,(1-T)}.$$

$$E\,[V^i] = \frac{T^2}{2\,(1-T)}, \text{ when } N \to \infty.$$

$$1 - T = \frac{T^2}{2\,(1-T)}, \text{ when } N \to \infty.$$

$$T = 2 - \sqrt{2} = 0.586.$$

A.2 Nonblocking Switch with Input Queueing/Output Queueing and Switch Speedup

This subsection derives the throughput of a nonblocking switch with input queueing/output queueing and a speedup factor 2 (i.e., S = 2). Follow the notations used in Section A.1. Since there are two switch slots in one link slot, the queueing equation can be derived using the switch slot as a unit. Define $V^i_{m-1;m-0.5}$ to be the number of packets in virtual queue i at the end of the (m-0.5)th link slot and $V^i_{m-0.5;m}$ to be the number of packets in virtual queue i at the end of the mth link slot. Define $B^i_{m-1;m-0.5}$ to be the number of arrival packets to virtual queue i during the (m-0.5)th link slot and $B^i_{m-0.5;m}$ to be the number of arrival packets to virtual queue i during the mth link slot. Define $F_{m-1;m-0.5}$ to be the number of packets left virtual queue i during the (m-0.5)th link slot and $F_{m-0.5;m}$ to be the number of packets left virtual queue i during the mth link slot. For the pure output queueing switch, the arrival process to the output queue is assumed to have a binomial distribution. In this case, the arrival process to the output queue is replaced by $F_{m-1;m-0.5}$ and $F_{m-0.5;m}$ for the mth link slot.

Following the procedure in Section A.1, it can be shown that the throughput is exactly 2 times larger than that of the pure input queueing. With a speedup factor S = 2, the maximum achievable throughput of the switch is approaching to 1.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1993	Final Contractor Report

4. TITLE AND SUBTITLE

On-Board B-ISDN Fast Packet Switching Architectures
Phase II: Development
Proof-of-Concept Architecture Definition Report

5. FUNDING NUMBERS

WU-506-72-21
C-NASW-4711

6. AUTHOR(S)

Dong-Jye Shyy and Wayne Redman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

COMSAT Laboratories
22399 COMSAT Drive
Clarksburg, Maryland 20871

8. PERFORMING ORGANIZATION REPORT NUMBER

E-7931

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR-191151

11. SUPPLEMENTARY NOTES

Project Manager, William D. Ivancic, (216) 433-3494.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 17

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

For the next-generation packet switched communications satellite system with on-board processing and spot-beam operation, a reliable on-board fast packet switch is essential to route packets from different uplink beams to different downlink beams. The rapid emergence of point-to-multipoint services such as video distribution, and the large demand for video conference, distributed data processing, and network management makes the multicast function essential to a fast packet switch (FPS). The satellite's inherent broadcast features gives the satellite network an advantage over the terrestrial network in providing multicast services. This report evaluates alternate multicast FPS architectures for on-board baseband switching applications and selects a candidate for subsequent breadboard development. Architecture evaluation and selection will be based on the study performed in Phase I, "On-Board B-ISDN Fast Packet Switching Architectures," and other switch architectures which have become commercially available as large scale integration (LSI) devices.

14. SUBJECT TERMS

Fast packet switch; B-ISDN; Congestion control

15. NUMBER OF PAGES
126

16. PRICE CODE
A07

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	